



PUBLIC

SAP Data Services

Document Version: 4.2 Support Package 13 (14.2.13.0) – 2021-07-27

Designer Guide

Content

1	Introduction to SAP Data Services.	17
2	Data protection and privacy.	18
3	Overview of this guide.	19
3.1	About this guide.	19
3.2	Who should read this guide.	19
3.3	Naming conventions and variables.	20
4	Designer user interface.	25
4.1	Project area.	26
4.2	Local object library.	27
4.3	Central Object Library.	28
	Central Object Library layout.	28
4.4	Viewing, hiding, and docking project area and object library.	29
4.5	Tool palette.	30
4.6	Workspace.	31
	Working with objects in the workspace.	32
	Scaling the workspace.	33
	Arrange workspace windows.	33
	Closing workspace windows.	34
4.7	General and environment options.	34
	Designer Environment options.	35
	Designer General options.	37
	Designer Graphics options.	39
	Designer Fonts options.	40
	Designer Attribute Values options.	41
	Designer Central Repository Connections options.	42
	Designer Language options.	42
	Designer SSL options.	43
	Data General options.	44
	Job Server Environment option.	44
	Job Server General options.	45
	Setting SAP environment options.	46
5	Logging in to the Designer.	48
5.1	Version restrictions.	49
5.2	Resetting users.	49

6	Reserved words.	50
7	Objects.	52
7.1	Reusable objects.	53
7.2	Single use objects.	54
7.3	Save reusable objects.	55
7.4	Object metadata.	57
7.5	Object descriptions.	58
7.6	Object hierarchy.	62
	Projects and subordinate objects.	63
	Work flows.	64
	Data flows.	64
7.7	Object naming conventions.	65
7.8	Object editors.	66
7.9	Creating a reusable object in the object library.	67
7.10	Creating a reusable object using the tool palette.	68
7.11	Adding an existing object.	69
7.12	Changing object names.	70
7.13	Adding, changing, and viewing object properties.	71
	Object Properties, General tab.	72
	Object Properties, Attributes tab.	73
	Object Properties, Class Attributes tab.	73
7.14	Annotations.	74
	Creating and deleting annotations.	75
7.15	Object descriptions.	75
	Adding a description to an object.	77
	Displaying a description in the workspace.	77
	Hiding object descriptions in the workspace.	77
	Editing object descriptions in the workspace.	77
7.16	Cutting or copying objects.	78
7.17	Replicating objects.	79
7.18	Save and delete objects.	80
	Saving changes to single reusable objects.	81
	Saving all changed objects in the repository.	81
	About deleting objects.	81
	Deleting an object definition from the repository.	82
	Deleting an object call.	83
7.19	Searching for objects.	83
8	Projects and Jobs.	86
8.1	Projects.	86
	Objects that make up a project.	86

	Creating a new project.	87
	Opening an existing project.	88
	Saving all changes to a project.	88
8.2	Jobs.	88
	Creating a job in the project area.	89
	Creating a job in the object library.	90
	Naming conventions for objects in jobs.	90
9	Datastores.	92
9.1	Database datastores.	93
	Defining a database datastore.	96
	Datastore metadata	102
	Imported metadata from database datastores.	107
	Attunity Connector (mainframe datastore).	115
	Data Federator	121
	IBM DB2 datastores.	124
	Informix.	127
	Microsoft SQL Server datastore.	129
	Memory datastores.	136
	MySQL datastore.	145
	ODBC	161
	Oracle datastores.	166
	Netezza.	170
	SAP SQL Anywhere.	172
	SAP ASE.	173
	Sybase IQ.	174
	Teradata	179
9.2	Persistent cache datastores.	181
	Defining a persistent cache datastore.	182
	Persistent Cache datastore options.	182
	Create persistent cache tables.	183
	Use persistent cache tables	186
9.3	Linked datastores.	186
	Relationship between database links and datastores.	187
	Linking a target datastore to a source datastore using a database link.	189
9.4	Adapter datastores.	190
9.5	Application datastores.	191
9.6	Web service datastores.	194
	Defining a web service datastore.	194
	Browse WSDL and WADL metadata through a web services datastore.	195
9.7	Change a datastore definition.	195
	Changing datastore options.	196

	Changing datastore properties.	196
9.8	Create and manage multiple datastore configurations.	197
	Multiple configuration datastore terminology.	198
	Why use multiple datastore configurations?.	199
	Creating a new datastore configuration.	199
	About working with Aliases.	202
	Creating an alias.	202
	Functions to identify the configuration.	203
	Datastore configuration in dataflows	204
	Portability solutions.	206
	Rename Owner option.	212
	Define a system configuration.	216
10	Flat file formats.	221
10.1	File format features.	222
	Reading multiple files at one time.	223
	Identifying source file names.	224
	Ignoring rows with specified markers.	224
	About special markers.	224
	Setting the number of threads for parallel processing.	225
	Error handling for flat file sources.	226
10.2	Creating a new flat file format template.	228
	Defining column attributes when creating a file format.	229
	Number formats.	230
	Date and time formats at the field level.	231
	Use a sample file to define column attributes.	232
	Defining column attributes using a sample file.	233
10.3	Replicating file formats.	234
10.4	Creating a file format from an existing flat table schema.	235
10.5	Creating a specific source or target file.	235
10.6	Editing file formats.	236
	Editing a source or target file.	237
	Changing multiple column properties.	238
10.7	File transfer protocols.	239
	Custom file transfers.	239
	Using a custom transfer program.	244
11	HDFS file format.	245
11.1	Configuring custom Pig script results as source	245
12	Working with COBOL copybook file formats.	247
12.1	Creating a new COBOL copybook file format.	247

12.2	Creating a new COBOL copybook file format and a data file.	248
12.3	Creating rules to identify which records represent which schemas.	249
12.4	Identifying the field that contains the length of the schema's record.	249
13	Creating Microsoft Excel workbook file formats on UNIX platforms.	250
13.1	Creating a Microsoft Excel workbook file format on UNIX.	250
14	Creating Web log file formats.	252
14.1	Word_ext function.	253
14.2	Concat_date_time function.	253
14.3	WL_GetKeyValue function.	254
15	Unstructured file formats.	255
16	File location objects.	256
16.1	Manage local and remote data files.	257
16.2	Creating a file location object.	259
16.3	Obtaining SSH authorization.	261
16.4	Editing an existing file location object.	261
16.5	Create multiple configurations of a file location object.	262
	Creating multiple file location object configurations.	262
16.6	Associate file location objects to file formats.	263
	Associating flat file format with file location object.	264
	Associating XML, DTD, or JSON schema files	265
	Associating COBOL copybooks with a file location object.	266
	Associating Excel workbooks with a file location object	267
16.7	Adding a file location object to data flow.	268
16.8	Using scripts to move files from or to a remote server.	268
	Moving files to and from Azure containers.	269
17	Data flows.	271
17.1	What is a data flow?.	271
	Naming data flows.	271
	Data flow example.	271
	Steps in a data flow.	272
	Data flows as steps in work flows.	272
	Intermediate data sets in a data flow.	273
	Operation codes.	273
	Passing parameters to data flows.	274
17.2	Creating and defining data flows.	275
	Defining a new data flow using the object library.	275
	Defining a new data flow using the tool palette.	275
	Changing properties of a data flow.	275

17.3	Source and target objects.	277
	Source objects.	278
	Target objects.	278
	Adding source or target objects to data flows.	282
	Template tables.	283
	Converting template tables to regular tables.	284
17.4	Understanding column propagation.	285
	Adding columns within a dataflow.	286
	Propagating columns in a data flow containing a Merge transform.	287
17.5	Lookup tables and the lookup_ext function.	288
	Accessing the lookup_ext editor.	289
	Example: Defining a simple lookup_ext function.	289
	Example: Defining a complex lookup_ext function.	292
17.6	Data flow execution.	294
	Push down operations to the database server.	294
	Distributed data flow execution.	295
	Load balancing.	296
	Caches.	297
17.7	Audit Data Flow overview.	297
18	Transforms.	299
18.1	Adding a transform to a data flow.	301
18.2	Transform editors.	302
18.3	Data Services Embedded Help.	302
18.4	Transform configurations.	304
	Creating a transform configuration.	304
	Adding a user-defined field.	305
	Ordered options editor.	306
	Associate, Match, and User-Defined transform editors.	306
18.5	Data assessment.	308
	Using the Data Profiler.	309
	Using View Data to determine data quality.	325
	Using the Validation transform.	327
	Using Auditing	335
18.6	Data Quality.	347
	Overview of data quality.	347
	Data Cleanse.	348
	Geocoding.	378
	Match.	400
	Address Cleanse.	483
	Beyond the basic address cleansing.	533
	Data quality statistics repository tables.	617

	Data Quality support for native data types.	618
	Data Quality data type definitions.	618
	Data Quality support for NULL values.	618
18.7	Query transform.	619
	Schema In and Schema Out.	620
	Query transform input schema.	621
	Query transform output schema.	623
	Output schema commands.	624
	Searching in an input or output schema.	626
18.8	Entity extraction.	627
	Use the Entity Extraction transform.	628
	Entities, types, subtypes, and facts.	629
	Differences between text data processing and data cleanse transforms.	630
19	Work flows.	632
19.1	Steps in a work flow.	632
19.2	Order of execution in work flows.	633
19.3	Example of a work flow.	634
19.4	Creating work flows.	634
	Creating a new work flow using the object library.	634
	Creating a new work flow using the tool palette.	635
	Specifying that a job executes the work flow one time.	635
	What is a single work flow?.	635
	What is a continuous work flow?.	636
19.5	Conditionals.	637
	Defining a conditional.	638
19.6	While loops.	639
	Defining a while loop.	640
	Using a while loop with View Data.	641
19.7	Try/catch blocks.	641
	Defining a try/catch block.	642
	Categories of available exceptions.	643
	Example: Catching details of an error.	644
	Catch best practices.	644
19.8	Scripts.	645
	Creating a script.	646
	Debugging scripts using the print function.	646
19.9	Smart Editor and function tools.	647
	Accessing the Smart Editor.	647
	Using the selection list and a tool tips.	648
	Browsing for a function.	649
	Searching for a function.	649

	Validating in Smart Editor.	650
20	Nested Data.	651
20.1	Representing hierarchical data.	651
20.2	Formatting XML documents.	653
	XML Schema specification.	654
	About importing XML schemas.	655
	Support for abstract datatypes.	663
	Importing substitution groups.	665
	Limiting the number of substitution groups to import.	665
	Specifying source options for XML files.	666
	Mapping optional schemas.	667
	Using Document Type Definitions (DTDs).	668
	Generating DTDs and XML Schemas from an NRDM schema.	673
20.3	Operations on nested data.	674
	Overview of nested data and the Query transform.	674
	FROM clause construction.	675
	Nesting columns.	678
	Using correlated columns in nested data.	679
	Distinct rows and nested data.	680
	Grouping values across nested schemas.	680
	Unnesting nested data.	681
	Transforming lower levels of nested data.	684
20.4	XML extraction and parsing for columns.	684
	Sample scenarios.	685
20.5	JSON extraction.	691
	Extracting data from JSON string using extract_from_json function.	691
21	Embedded Data Flows.	693
21.1	Overview of embedded data flows.	693
21.2	Embedded data flow examples.	693
21.3	Creating embedded data flows.	694
	Using the Make Embedded Data Flow option.	695
	Creating embedded data flows from existing flows.	696
	Using embedded data flows.	697
	Separately testing an embedded data flow.	699
	Troubleshooting embedded data flows.	699
22	Variables and Parameters.	701
22.1	Viewing variables and parameters.	703
22.2	Variables and Parameters dialog box.	703
22.3	Definitions and Calls tabs.	704

22.4	Object types and uses for variables and parameters.	705
22.5	Accessing View Where Used for variables and parameters.	706
22.6	Use local variables and parameters.	706
	Parameters.	707
	Adding a parameter to a work flow or data flow.	707
	Pass values into data flows.	708
	Setting the value of the parameter in the flow call.	709
	Defining a local variable.	710
	Replicating a local variable.	710
22.7	Use global variables	711
	Defining a global variable.	711
	Replicating a global variable.	712
	Viewing global variables from the Properties dialog.	712
	Values for global variables.	713
	Setting a global variable value as a job property.	713
	Setting a global variable value as an execution property.	714
	Automatic ranking of global variable values in a job.	714
22.8	Local and global variable rules.	717
22.9	Environment variables.	717
22.10	Set file names at run-time using variables.	718
	Using a variable in a flat file name.	718
22.11	Substitution parameters.	720
	Overview of substitution parameters.	720
	Tasks in the Substitution Parameter Editor.	722
	Substitution parameter configurations and system configurations.	726
	Associating substitution parameter configurations with system configurations.	726
	Overriding a substitution parameter in the Administrator.	728
	Executing a job with substitution parameters	728
	Exporting and importing substitution parameters.	729
23	Real-time Jobs.	730
23.1	Request-response message processing.	730
23.2	What is a real-time job?.	731
	Content of real-time job.	731
	Real-time versus batch.	732
	Messages.	732
	Message processing.	734
	Real-time job examples.	734
23.3	Creating real-time jobs.	736
	Arrange metadata	736
	Real-time job models.	736
	Using real-time job models.	738

	Creating a real-time job with a single data flow.	740
23.4	Real-time source and target objects.	741
	Viewing an XML message source or target schema.	742
	Load targets as a single transaction.	742
	Secondary sources and targets.	742
	Transactional loading of tables.	743
	Design tips for data flows in real-time jobs.	744
23.5	Testing real-time jobs.	744
	Executing a real-time job in test mode.	744
	Using View Data.	745
	Using an XML file target.	745
23.6	Building blocks for real-time jobs.	746
	Supplementing message data.	746
	Branching data flow based on a data cache value.	749
	Calling application functions.	750
23.7	Designing real-time applications.	750
	Reducing queries requiring back-office application access.	750
	Messages from real-time jobs to adapter instances.	751
	Real-time service invoked by an adapter instance.	751
	Start and stop real-time services.	751
24	Executing Jobs.	753
24.1	Overview of job execution.	753
24.2	Preparing for job execution.	754
	Validate jobs and job components.	754
	Ensuring that the Job Server is running.	755
	Setting job execution options.	756
24.3	Executing jobs as immediate tasks.	759
	Executing a job as an immediate task.	759
	Monitor tab	760
	Log tab	760
24.4	Debugging execution errors.	761
	Access logs.	761
	Trace properties.	764
	Examining target data.	767
	Creating a debug package.	767
24.5	Changing Job Server options.	768
	Changing option values for an individual Job Server.	770
	Using mapped drive names in a path.	772
25	Functions and Procedures.	773
25.1	About functions.	773

	Functions compared with transforms.	774
	Operation of a function.	774
	Arithmetic in date functions.	775
	Including functions in expressions.	775
	Custom functions.	778
	Built-in functions.	783
25.2	About procedures.	787
	Before you use procedures.	788
	Creating stored procedures in a database.	789
	Importing metadata for stored procedures	794
	Structure of a stored procedure.	795
	Calling stored procedures.	796
	Checking execution status	802
26	Scripting Language.	803
26.1	Using the scripting language.	803
26.2	Language syntax.	804
	Syntax for statements in scripts.	805
	Syntax for column and table references in expressions.	805
	Strings.	806
	Variables.	807
	Variable interpolation.	808
	Functions and stored procedures.	809
	NULL values.	809
	Validate.	811
26.3	Sample scripts.	812
	Square function.	813
	RepeatString function.	813
27	Design and Debug.	815
27.1	Using View Where Used.	815
	Accessing View Where Used from the object library.	816
	Accessing View Where Used from the workspace.	818
	Limitations.	819
27.2	Using View Data.	819
	Accessing View Data.	820
	Viewing data in the workspace.	821
	View Data Properties.	823
	View Data tool bar options.	826
	View Data tabs.	827
27.3	Using the Design-Time Data Viewer.	831
	Viewing Design-Time Data.	831

	Configuring the Design-Time Data Viewer.	832
	Specifying variables for expressions.	832
27.4	Using the interactive debugger.	832
	Before starting the interactive debugger.	833
	Starting and stopping the interactive debugger.	836
	Panes.	837
	Debug menu options and tool bar.	842
	Viewing data passed by transforms.	844
	Push-down optimizer.	845
	Limitations.	845
27.5	Comparing Objects.	846
	Comparing two different objects.	846
	Comparing two versions of the same object.	846
	Overview of the Difference Viewer window.	847
	Navigating through differences.	851
27.6	Calculating column mappings.	851
	Automatically calculating column mappings.	852
	Manually calculating column mappings.	852
27.7	Bypassing specific work flows and data flows.	852
	Bypassing a single data flow or work flow.	853
	Bypassing multiple data flows or work flows.	853
	Disabling bypass.	854
28	Recovery Mechanisms.	855
28.1	Recovering from unsuccessful job execution.	855
28.2	Automatically recovering jobs.	856
	Enabling automated recovery.	856
	Marking recovery units.	856
	Running in recovery mode.	857
	Ensuring proper execution path.	858
	Using try/catch blocks with automatic recovery.	859
	Ensuring that data is not duplicated in targets.	861
	Using preload SQL to allow re-executable data flows	861
28.3	Manually recovering jobs using status tables.	863
28.4	Processing data with problems.	864
	Using overflow files.	864
	Filtering missing or bad values	865
	Handling facts with missing dimensions.	865
28.5	Exchanging metadata.	866
	Metadata exchange.	866
28.6	Loading Big Data file with recovery option.	868
	Turning on the recovery option for Big Data loading.	868

	Limitations.	869
29	Changed Data capture.	871
29.1	Full refresh.	871
29.2	Capture only changes.	872
29.3	Source-based and target-based CDC.	872
	Source-based CDC.	872
	Target-based CDC.	873
29.4	Use CDC with Oracle sources.	874
	Overview of CDC for Oracle databases.	874
	Set up Oracle CDC.	879
	Creating a CDC datastore for Oracle.	880
	Import CDC data into tables.	881
	Viewing an imported CDC table.	884
	Configuring an Oracle CDC source table.	886
	Creating a data flow with an Oracle CDC source.	888
	Maintaining CDC tables and subscriptions.	889
	Limitations.	890
29.5	Use CDC with Attunity mainframe sources.	890
	Setting up Attunity CDC.	891
	Setting up the software for CDC on mainframe sources.	892
	Importing mainframe CDC data.	894
	Configuring a mainframe CDC source.	895
	Using mainframe check-points.	896
	Limitations.	898
29.6	Use CDC with SAP Replication Server.	898
	Overview for using a continuous work flow and functions.	899
	Overview for using the SAP PowerDesigner modeling method.	904
29.7	Use CDC with Microsoft SQL Server databases.	914
	Limitations.	916
	Data Services columns.	917
	Changed Data Capture (CDC) method.	919
	Change Tracking method.	923
	Replication Server method.	926
29.8	Use CDC with timestamp-based sources.	935
	Processing timestamps.	936
	Overlaps.	938
	Types of timestamps.	943
	Timestamp-based CDC examples.	944
	Additional job design tips.	949
29.9	Use CDC for targets.	950

30	Data Services data types.	951
31	Data type conversion.	954
31.1	Date arithmetic.	954
31.2	Unsupported data types.	955
31.3	Conversion to or from internal data types.	956
	Attunity Streams.	958
	Cobol copybook.	959
	Data Federator.	960
	IBM DB2.	961
	Informix.	963
	Microsoft Excel.	965
	Microsoft SQL Server.	966
	MySQL.	969
	Netezza.	971
	ODBC.	972
	Oracle.	974
	SQL Anywhere.	976
	SAP ASE.	979
	SAP Sybase IQ.	980
	Teradata.	982
31.4	Conversion of data types within expressions.	984
31.5	Conversion of number data types in expressions.	985
	Division algorithm.	986
	Conversion of decimals with different scale and precision.	986
	Conversion between string and number data types	987
	Conversion between Oracle string and date data types.	987
31.6	Conversion between explicit data types.	988
31.7	Conversion between native data types.	988
	date.	989
	datetime.	990
	decimal.	991
	double.	991
	int (integer).	993
	varchar.	994
32	Monitor jobs in the Administrator.	995
33	Multi user Development.	996
33.1	Central versus local repository.	996
33.2	Multiple users.	998
33.3	Security and the central repository.	1000

33.4	Multi-user Environment Setup.	1001
	Creating a nonsecure central repository.	1001
	Defining a connection to a nonsecure central repository.	1002
	Activate a central repository.	1003
33.5	Implementing Central Repository Security.	1005
	Overview.	1005
	Creating a secure central repository.	1006
	Adding a multi-user administrator (optional).	1007
	Setting up groups and users.	1008
	Defining a connection to a secure central repository.	1008
	Working with objects in a secure central repository.	1009
33.6	Working in a Multi-user Environment.	1010
	Filtering.	1010
	Adding objects to the central repository.	1011
	Checking out objects.	1013
	Undoing check out.	1016
	Checking in objects.	1017
	Labeling objects.	1019
	Getting objects.	1022
	Comparing objects.	1023
	Viewing object history.	1024
	Deleting objects.	1026
33.7	Migrating Multi-user Jobs.	1027
	Application phase management.	1027
	Copying contents between central repositories.	1029
	Central repository migration.	1030

1 Introduction to SAP Data Services

SAP Data Services delivers a single enterprise-class solution for data integration, data quality, data profiling, and text data processing.

Businesses can use Data Services to integrate, transform, improve, and deliver trusted data to critical business processes. IT organizations can depend on Data Services for maximum operational efficiency to improve data quality and gain access to heterogeneous sources and applications. Data Services provides all of these features using:

- Single development user interface
- Metadata repository
- Data connectivity layer
- Runtime environment
- Management Console

2 Data protection and privacy

SAP provides specific features and functions to support compliance with regard to relevant legal requirements, including data protection and privacy.

Data protection is associated with numerous legal requirements and privacy concerns. In addition to compliance with applicable data privacy regulations, SAP needs to consider compliance with industry-specific legislation in different countries. SAP does not give any advice on whether the provided features and functions are the best method to support company, industry, regional, or country-specific requirements. Furthermore, SAP-provided information does not give any advice or recommendation in regards to additional features that are required in particular IT environments. Make your decisions related to data protection on a case-by-case basis, considering your given system landscape and applicable legal requirements.

i Note

In the majority of cases, SAP compliance with applicable data protection and privacy laws are not covered by a product feature. SAP software supports data protection compliance by providing security features and specific data protection-relevant functions, such as simplified blocking and deletion of personal data. SAP does not provide legal advice in any form. SAP uses definitions and other terms in this document that are not from any given legal source.

Be aware that SAP software places the data that you provide into trace logs, sample reports, and repositories (side-effect data), and so on. In other words, your data finds its way into places other than output files. It is your responsibility to delete this data.

In addition, the Data Services application does not run any virus scan applications on persisted data. To protect persisted data we strongly recommend running your own virus scan software on any system that runs Data Services.

Here are some examples of where Data Services uses customer data:

- When you enable the *Trace all* option, Data Services prints data in some log files.
- If a job using bulk load functionality fails, Data Services saves data files that contain customer data in the Bulkload directory so you can review and analyze the data. The default bulk load location is `%DS_COMMON_DIR%/log/BulkLoader`.
- The `smtp_to` and `mailer` functions use mail ID numbers as input parameters. Data Services saves mail ID numbers in the Data Services repository.
- Data Services places a sampling of customer data in the Data Services repository when the “full” option is enabled during side-effect data generation for Global Address Cleanse (GAC) and Universal Data Cleanse (UDC).
- The GAC sample report contains a sample of user address data.

3 Overview of this guide

Welcome to the SAP Data Services Designer Guide. This guide describes all aspects of the Data Services graphical user interface (GUI) that we call the Designer.

The Designer offers a rich GUI development environment in which to define your data application logic. Use the many features to manipulate various data types from traditional database management systems, big data, and cloud data.

[About this guide \[page 19\]](#)

The Designer Guide contains both conceptual and procedural information for SAP Data Services Designer.

[Who should read this guide \[page 19\]](#)

Any user of SAP Data Services Designer should use the Designer Guide as a source of information about software features and procedures.

[Naming conventions and variables \[page 20\]](#)

Familiarize yourself with how we use SAP and SAP Data Services system names and specific environment variables throughout this guide.

3.1 About this guide

The Designer Guide contains both conceptual and procedural information for SAP Data Services Designer.

Conceptual information helps you understand Designer and how it works. Procedural information provides steps for accomplishing tasks with Designer.

The *Designer Guide* is most useful to you for the following activities:

- When you learn about the product.
- When you begin to use the software for frequent tasks in all phases of your data-movement projects.
- When you perform infrequent tasks or use unfamiliar features in all phase of your projects.

As you become familiar with the software and tools, you may become less dependent on the *Designer Guide*. However, when you perform infrequent tasks, or you try a feature with which you are unfamiliar, the *Designer Guide* becomes a handy source of information.

3.2 Who should read this guide

Any user of SAP Data Services Designer should use the Designer Guide as a source of information about software features and procedures.

This and other Data Services product documentation assumes the following about the users of Data Services Designer:

- You are an application developer, consultant, or database administrator working on data extraction, data warehousing, data integration, or data quality.
- You understand your source data systems, RDBMS, business intelligence, and messaging concepts.
- You understand the data needs of your organization.
- You are familiar with SQL.
- You are familiar with the Data Services installation environments—Microsoft Windows or UNIX.
- You are familiar with networks and computer system architecture.

In addition, if you use Data Services to design jobs for real-time processing, we expect that you are familiar with the following concepts:

- DTD and XML Schema formats for XML files.
- Publishing Web Services such as WSDL, REST, HTTP, and SOAP protocols.

3.3 Naming conventions and variables

Familiarize yourself with how we use SAP and SAP Data Services system names and specific environment variables throughout this guide.

Shortened names

When we refer to locations for SAP and SAP Data Services files, we refer to certain systems with shortened names plus we use specific variables and environment variables.

Terminology

- The terms “Data Services system” and “SAP Data Services” mean the same thing.
- The term “BI platform” refers to the SAP BusinessObjects Business Intelligence platform.
- The term “IPS” refers to SAP BusinessObjects Information platform services.

i Note

Data Services requires BI platform components. However, when you don't use other SAP applications, IPS, a scaled back version of BI, also provides these components for Data Services.

- “CMC” refers to the Central Management Console provided by the BI platform or IPS.
- “CMS” refers to the Central Management Server provided by the BI platform or IPS.

Variables

The following table describes the various variables and environment variables that we use throughout our documentation.

Variables	Description
INSTALL_DIR	The installation directory for the SAP software. Default location: - For Windows: C:\Program Files (x86)\SAP BusinessObjects - For UNIX: \$HOME/sap businessobjects i Note INSTALL_DIR isn't an environment variable. The installation location of SAP software can be different than what we list for INSTALL_DIR based on the location that your administrator sets during installation.
BIP_INSTALL_DIR	The directory for the BI platform or IPS. Default location: - For Windows: <INSTALL_DIR>\SAP BusinessObjects Enterprise XI 4.0 ❖ Example<pre>C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0</pre> - For UNIX: <INSTALL_DIR>/enterprise_xi40 i Note These paths are the same for both BI platform and IPS. i Note BIP_INSTALL_DIR isn't an environment variable. The installation location of SAP software can be different than what we list for BIP_INSTALL_DIR based on the location that your administrator sets during installation.

Variables	Description
<LINK_DIR>	An environment variable for the root directory of the Data Services system. Default location: All platforms <INSTALL_DIR>\Data Services ❖ Example C:\Program Files (x86)\SAP BusinessObjects\Data Services

Variables	Description
<DS_COMMON_DIR>	An environment variable for the common configuration directory for the Data Services system. Default location: - If your system is on Windows (Vista and newer): <AllUsersProfile>\SAP BusinessObjects Data Services Note The default value of <AllUsersProfile> environment variable for Windows Vista and newer is C:\ProgramData. Example C:\ProgramData\SAP BusinessObjects Data Services - If your system is on Windows (Older versions such as XP) <AllUsersProfile>\Application Data \SAP BusinessObjects\Data Services Note The default value of <AllUsersProfile> environment variable for Windows older versions is C:\Documents and Settings\All Users. Example C:\Documents and Settings\All Users\Application Data\SAP BusinessObjects\Data Services - UNIX systems (for compatibility) <LINK_DIR> The installer automatically creates this system environment variable during installation. Note Starting with Data Services 4.2 SP6, users can designate a different default location for <DS_COMMON_DIR> during installation. If you can't find the <DS_COMMON_DIR> in the listed default location, ask

Variables	Description
	your system administrator to find out where your default location is for <code><DS_COMMON_DIR></code>.
<code><DS_USER_DIR></code>	The environment variable for the user-specific configuration directory for the Data Services system. Default location: - If you're on Windows (Vista and newer): <code><UserProfile>\AppData\Local\SAP BusinessObjects\Data Services</code> i Note The default value of <code><UserProfile></code> environment variable for Windows Vista and newer versions is <code>C:\Users\{username}</code>. - If you're on Windows (Older versions such as XP): <code><UserProfile>\Local Settings \Application Data\SAP BusinessObjects \Data Services</code> i Note The default value of <code><UserProfile></code> environment variable for Windows older versions is <code>C:\Documents and Settings\{username}</code>. i Note The system uses <code><DS_USER_DIR></code> only for Data Services client applications on Windows. UNIX platforms don't use <code><DS_USER_DIR></code>. The installer automatically creates this system environment variable during installation.

4 Designer user interface

The SAP Data Services Designer user interface provides the tools for you to create objects such as projects, jobs, work flows, and data flows, and to test, debug, and execute jobs.

Designer contains industry standard GUI elements such as menu and tool bars. Each element has an identifying label or a tool tip that help you learn the elements of the menu options and tools available in Designer. Additionally, Designer accesses objects in your Data Services repository through the object library.

When you open Designer in the default layout, you see the following screen elements:

- Menu bar across the top
- Tool bar under the menu bar
- Project Area in a dockable window at left that contains three tabs: *Designer*, *Monitor*, and *Log*
- Local Object Library in a dockable window at left that contains the following tabs:
 - *Projects*
 - *Jobs*
 - *Work flows*
 - *Data flows*
 - *Transforms*
 - *Datastores*
 - *Formats*
 - *Custom Functions*
- Workspace in the main pane
- Tool pallet at the far right

We refer to these areas in our conceptual and procedural topics. Spend some time clicking the various options and areas to get to know the layout of Designer.

To practice using Designer and to learn more about the tools and objects in Designer, see the *Tutorial*. View the tutorial on the SAP Help Portal at https://help.sap.com/viewer/product/SAP_DATA_SERVICES/.

Project area [page 26]

Use the project area pane to create and manage projects, to monitor job executions, and to view job execution logs.

Local object library [page 27]

The local object library provides access to reusable objects from the local repository.

Central Object Library [page 28]

The central object library acts as source control for managing changes to objects in an environment with multiple users.

Viewing, hiding, and docking project area and object library [page 29]

View, hide, or auto hide the project area and object library, or move the panes to new locations on the Designer screen.

Tool palette [page 30]

Use the tool palette to add new objects to the workspace.

Workspace [page 31]

The workspace is where you manipulate system objects and graphically assemble data movement processes.

[General and environment options \[page 34\]](#)

Use the general and environment options to set default appearance, fonts, language, and so on.

4.1 Project area

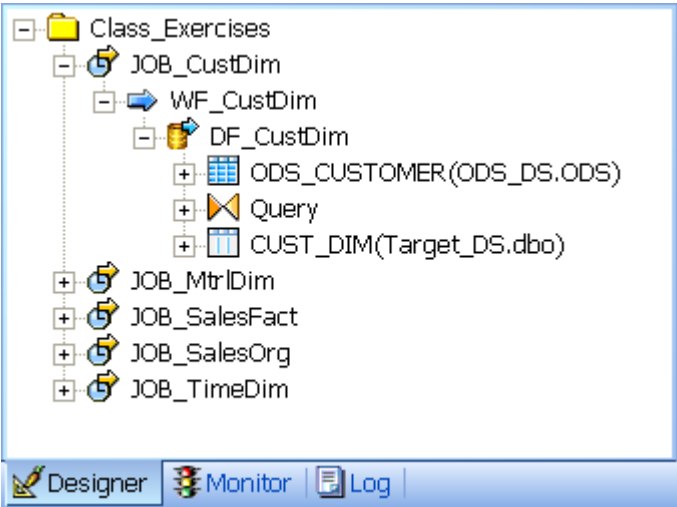
Use the project area pane to create and manage projects, to monitor job executions, and to view job execution logs.

By default the project area resides on the left side of Designer in a docked pane. You can hide the project area or move it to any other location.

The project area contains three tabs. The following table describes the purpose for each tab.

Tab	Description
 <i>Designer</i>	Manage existing projects and create new projects. Expand an existing project node to view a hierarchical list of all objects in the project. Open any object in the project to display in the workspace.
 <i>Monitor</i>	View the status of a currently executing job.
 <i>Log</i>	View the processing log of completed jobs.

The following screen capture shows the *Designer* tab of the project area. The project name is Class_Exercises. The software lists the underlying objects in a specific hierarchical order: Project, job, work flow, data flow.



4.2 Local object library

The local object library provides access to reusable objects from the local repository.

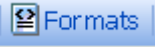
By default the object library resides on the left side of Designer in a docked pane under the project area. The top of the pane displays the name of the CMC repository and the current user in parentheses. You can hide the object library or move it to any other location.

The objects in the object library include built-in system objects such as transforms, and the objects you create, configure, and save such as datastores, jobs, data flows, and work flows.

The local object library is a window into your local repository. Accessing objects from the object library eliminates the need to access the repository directly. SAP Data Services Designer updates the repository through normal operations. For example, when you save a new object, Data Services adds the object to the repository. When you edit an existing object, Data Services updates the object in the repository.

The following table describes each tab in the object library.

Tab	Description
 Projects <i>Projects</i>	Contains all existing projects. Projects are sets of jobs that include other objects such as work flows, data flows, and transforms.
 Jobs <i>Jobs</i>	Contains all existing jobs. Jobs are executable work flows. There are two job types: batch jobs and real-time jobs.
 Work Flows <i>Work Flows</i>	Contains all existing work flows. Work flows order data flows and the operations that support data flows, defining the interdependencies between them.
 Data Flows <i>Data Flows</i>	Contains all existing data flows. Data flows tell the software how to process a task.
 Transforms <i>Transforms</i>	Contains software-supplied transform templates and custom transforms. Transforms operate on data, producing output data sets from the sources you specify.
 Datastores <i>Datastores</i>	Contains all existing datastores. Datastores represent connections to databases and applications used in your project. Expand a datastore node to view related objects such as functions, tables, template tables, and so on.

Tab	Description
 <i>Formats</i>	Contains all existing formats and file location objects. Formats describe the structure of a specific file type such as flat files, nested schemas, Excel workbooks, and so on. File location objects define properties for specific file transfer protocols.
 <i>Custom Functions</i>	Contains all existing custom functions. Create custom functions using Data Services own scripting language.

4.3 Central Object Library

The central object library acts as source control for managing changes to objects in an environment with multiple users.

Display the central object library in Designer after you create it. The central object library is a dockable and movable pane just like the project area and object library.

Through the central object library, authorized users access a library repository that contains versions of objects saved there from their local repositories. The central object library enables administrators to manage who can add, view and modify the objects stored in the central repository.

Users must belong to a user group that has permission to perform tasks in the central object library. Administrators can assign permissions to an entire group of users as well as assign various levels of permissions to the users in a group.

❖ Example

All users in Group A can get objects from the central object library. Getting an object means you place a copy of the object in your local repository. If the object exists in your local repository, Data Services updates your local copy with the most recent changes. User01 in Group A has administrator rights and can add, check in, edit, and check out objects. Plus User01 can set other user permissions for the objects.

In Designer, users check out an object from the central repository using the central object library. Once checked out, no other user can work on that object until it is checked back into the central repository. Other users can change their local copy of the object, but that does not affect the version in the central repository.

4.3.1 Central Object Library layout

The information in the central object library is different from the local object library.

In the central object library, there are several icons located at the top of the pane to perform the following tasks:

- Select various object checkout types
- Show history
- Edit central repository connection
- Refresh the content of the central object library

The top of the pane also displays the current user group to which you belong and the name of the central repository.

The main area of the central object library contains:

- A list of the object types based on the lower tab that you choose.
- When you check out an object, a red check mark appears over the object name in the left column.

The main area also contains several columns with information for each object:

- *Check out user*: The name of the user who currently has the object checked out of the library, or blank when the object is not checked out.
- *Check out repository*: The name of the repository that contains the checked-out object, or blank when the object is not checked out.
- *Permission*: The authorization type for the group that appears in the *Group Permission* box at the top. When you add a new object to the central object library, the current group gets FULL permission to the object and all other groups get READ permission.
- *Latest version*: A version number and a timestamp that indicate when the software saved this version of the object.
- *Description*: Information about the object that you entered when you added the object to the library.

4.4 Viewing, hiding, and docking project area and object library

View, hide, or auto hide the project area and object library, or move the panes to new locations on the Designer screen.

The project area and object library panes appear in the left side of Designer by default. Perform the following steps to hide, auto hide, view, or dock and undock the panes:

1. To hide a pane:
 - a. Right-click the gray border around the pane.
 - b. Select *Hide* from the popup menu.
2. To auto hide a pane:
 - a. Right-click the gray border around the pane.
 - b. Select *Auto Hide* from the popup menu.

The software moves the pane header to the nearest border of Designer. Click the header to unhide the pane. Click the push pin icon in the upper right corner of the pane to rehide.

3. To view the panes:
 - a. Select  *Tools* > *Project Area*. 
 - b. Select  *Tools* > *Object Library*. 

4. To undock, move, and dock the panes:
 - a. Click the gray border around the pane and drag the pane.
 - b. Drag the pane until your mouse pointer reaches one of the positional arrow icons that appear as you drag the pane.

Positional arrows appear on screen while you drag the pane. Each arrow indicates a position on the Designer screen to automatically dock the pane to that position.
 - c. Double-click the pane border to move to the last position whether it is a docked or undocked position.

Play around with the panes to learn the various ways to hide, unhide, and dock the panes. To go back to the default layout, select **View > Apply Default Layout.**

4.5 Tool palette

Use the tool palette to add new objects to the workspace.

The tool palette is a separate pane that appears by default on the right edge of the Designer workspace. Drag and drop the tool palette anywhere on your screen or drag it to the left, top, or right side and it automatically docks itself.

If the object in the tool palette is not applicable to the current content of your workspace, the software disables the icon. To show the name of an active icon, hold the cursor over the icon until the tool tip for the icon appears.

When you create an object from the tool palette, you create a new definition of an object. If a new object is reusable, the software makes it available in the object library after you create and save it. For example, if you add a data flow object from the tool palette to a job in the workspace and save the job, the software adds the data flow to the [Data Flows](#) tab in the object library.

The following table describes each object in the tool palette, whether it is reusable, and where you can use it.

Tool	Description and class	Available
 <i>Pointer</i>	Returns the tool pointer to a selection pointer for selecting and moving objects in a diagram.	Everywhere
 <i>Work flow</i>	Creates a new work flow. Reusable.	Jobs and work flows
 <i>Data flow</i>	Creates a new data flow. Reusable.	Jobs and work flows
 <i>ABAP data flow</i>	Creates a new ABAP data flow. Reusable.	SAP application
 <i>Query transform</i>	Creates a template for a query. Use it to define column mappings and row selections. Single use.	Data flows
 <i>Template table</i>	Creates a table for a target. Single use.	Data flows
 <i>Nested schema template</i>	Creates a JSON or XML template. Single use.	Data flows

Tool	Description and class	Available
 <i>Data transport</i>	Required when you create an ABAP data flow. Single use.	SAP application
 <i>Script</i>	Creates a new script object. Single use.	Jobs and work flows
 <i>Conditional</i>	Creates a new conditional object. Single use.	Jobs and work flows
 <i>Try</i>	Creates a new try object. Single-use.	Jobs and work flows
 <i>Catch</i>	Creates a new catch object. Single use.	Jobs and work flows
 <i>Annotation</i>	Creates an annotation. Single use.	Jobs, work flows, and data flows

For complete descriptions of each of the objects, see the *Reference Guide*.

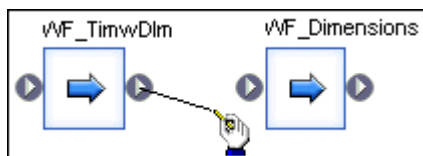
4.6 Workspace

The workspace is where you manipulate system objects and graphically assemble data movement processes.

When you open or select a job or any flow within a job hierarchy, the workspace becomes "active" with your selection. The workspace shows the objects and flow of data in each job or flow you open.

Drag and drop various icons onto the workspace and connect each object using your mouse pointer to create a diagram that represents a process. You specify the flow of data by connecting objects from left to right in the order you want the data to move. The resulting diagram is a visual representation of an entire data movement application or some part of a data movement application.

For example, the following diagram shows two work flows being connected by a user. Each work flow opens to reveal more processes.



Each object that you add to the workspace has connection indicators on the right, left, or both sides. For example, the connection indicators on the two work flows in the diagram are arrows. Connection indicators on objects such as sources, transforms, and targets appear as small square knobs. Not all objects contain a connector on both right and left sides. For example, a source object contains a connector on the right side only. A target object contains a connector on the left side only.

[Working with objects in the workspace \[page 32\]](#)

Use standard mouse controls to work with objects in the workspace.

[Scaling the workspace \[page 33\]](#)

By scaling the workspace, you can change the focus of a job, work flow, or data flow.

[Arrange workspace windows \[page 33\]](#)

Use the *Window* menu to arrange multiple open workspace windows in a similar manner as Microsoft Windows window arrangements.

[Closing workspace windows \[page 34\]](#)

Close workspace views that you are not using to save system resources.

4.6.1 Working with objects in the workspace

Use standard mouse controls to work with objects in the workspace.

1. To add objects to the workspace:
 - a. With an applicable process open in the workspace, select an object from the object library.
 - b. Drag and drop the object onto the workspace in the applicable location.
 - c. Rename the object as applicable.
2. To add objects from the tool pallet:
 - a. Click the object in the tool palette to select it.
 - b. Click an empty area of the workspace. The software adds the object to the workspace.
 - c. Drag and drop the object to the applicable location in the workspace.
 - d. Rename the object as applicable.
3. To connect objects in the workspace:
 - a. Click the square knob on the right side of the first object and drag your mouse to the square knob on the left side of the second object to connect. Your mouse draws a line that connects the two objects.

You can draw a connection line only from one connection indicator to another. The objects in a data flow contain square knobs as the connection indicators. When you connect two work flows, the work flow icons have arrows for connection indicators. The connection indicators appear on the left or right side of the object, or on both sides of the object.
 - b. Continue connecting all objects on the workspace in the order in which you want the software to process the objects.
4. To disconnect objects in the workspace:
 - a. Click the connecting line of the objects to disconnect.
 - b. Press

The software still considers the object as in use. For example, when you check the transform to see where it is used, the software lists the data flow as using the transform, even though the transform is not connected to the flow.
5. To delete an object in the workspace:
 - a. Select the object to be deleted.
 - b. Press

Deleting an object also deletes the call to the object.
6. To arrange items in the workspace:
 - a. Right-click in the workspace.
 - b. Select *Auto Layout*.

The software arranges your connected objects in a straight line.

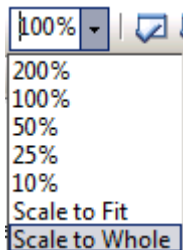
4.6.2 Scaling the workspace

By scaling the workspace, you can change the focus of a job, work flow, or data flow.

A workspace can become full of objects based on how complicated the project is. Scaling or resizing the workspace enables you to examine a particular portion of a data flow, for example. Reduce the scale to examine the entire work flow without scrolling.

1. Select the dropdown arrow on the scale icon in the toolbar. Alternately, right-click in the workspace.

The software displays the following dropdown menu:



2. Select a specific percentage from the dropdown list to enlarge or reduce the view.
3. Select *Scale to Fit* from the dropdown list to fit the entire project in your current view area.
4. Select *Scale to Whole* from the dropdown list to show the entire workspace area in the current view area.

4.6.3 Arrange workspace windows

Use the *Window* menu to arrange multiple open workspace windows in a similar manner as Microsoft Windows window arrangements.

Based on the number of processes you have open, you may have several workspaces open but you can view only one at a time. Arrange the windows so that you can better visualize what you have open.

The default layout is with one object filling the workspace pane. When you have more than one object open, the default layout shows tabs at the bottom of the pane. Move from one object to another object by clicking the applicable tab.

i Note

You can hide the tabs by making a setting in the general and environment options.

To change the arrangement of panes in the workspace, select *Window* and one of the menu options described in the following table.

Option	Description
<i>Cascade</i>	Displays each workspace in a cascading manner. The software arranges the workspaces so that each workspace title is visible, but only one workspace is in full view. Click the header of each workspace to bring that workspace into focus.
<i>Tile Horizontally</i>	Displays each workspace in a horizontal manner so that you can see a smaller version of each workspace in your view area.
<i>Tile Vertically</i>	Displays each workspace in a vertical manner so that you can see a smaller version of each workspace in your view area.

Related Information

[General and environment options \[page 34\]](#)

4.6.4 Closing workspace windows

Close workspace views that you are not using to save system resources.

When you drill into an object in the project area or workspace, each object that you select opens a view in the workspace. Because multiple views open in your workspace use up system resources and may impact performance, it is important to close the views you do not need.

Perform the following steps to close views that you are not using:

1. Determine if you have multiple views open in your workspace area by looking at the tabs at the bottom of the workspace area, or by arranging the views using the [Windows](#) menu.
2. Click the **X** in the upper right corner of each window to close the views that you do not need open.
3. Alternately, close all views by selecting  [Window](#)  [Close All Windows](#)  or by selecting the [Close All Windows](#) icon in the toolbar.

4.7 General and environment options

Use the general and environment options to set default appearance, fonts, language, and so on.

Make general and environment settings in the [Options](#) dialog box. To open the [Options](#) dialog box, select  [Tools](#)  [Options](#) .

Expand each node in the *Options* dialog box to select the type of options to set. When you select and expand each option in the *Category* pane, the software displays options and descriptions on the right.

[Designer Environment options \[page 35\]](#)

With the environment options, change default settings for metadata reporting and Job Servers, as well as communication port settings for Designer.

[Designer General options \[page 37\]](#)

Set options in the *General* group to define default settings for commonly used options in Designer.

[Designer Graphics options \[page 39\]](#)

Use settings in the *Graphics* options to choose and preview stylistic elements to customize your workspaces.

[Designer Fonts options \[page 40\]](#)

Use the *Font* options to set font type, style, and size for text in the workspace pane.

[Designer Attribute Values options \[page 41\]](#)

Set *Attribute Values* options to preserve or clear existing attribute values for columns or tables when you re import data.

[Designer Central Repository Connections options \[page 42\]](#)

Use the *Central Repository Connections* options to define default connection settings for the central repository.

[Designer Language options \[page 42\]](#)

Set *Language* options to set a locale other than the default locale of English.

[Designer SSL options \[page 43\]](#)

Specify options in SSL to change the default SSL certificate and key file settings established when your administrator installed the software.

[Data General options \[page 44\]](#)

Use the options in *General* to set default data options in the software.

[Job Server Environment option \[page 44\]](#)

Set the *Environment* option to control the number of engine processes that SAP Data Services performs in parallel.

[Job Server General options \[page 45\]](#)

Set *General* options to change default option values for an individual Job Server.

[Setting SAP environment options \[page 46\]](#)

Set *Environment* options in SAP Data Services for processes that are specific to SAP applications.

4.7.1 Designer Environment options

With the environment options, change default settings for metadata reporting and Job Servers, as well as communication port settings for Designer.

Default Administrator for Metadata Reporting

Option	Description
<i>Administrator</i>	Specifies the host computer that the software uses for metadata reporting. The software defines the default administrator by <code>host_name:port_number</code>. Select the applicable option from the dropdown list. The software lists only the hosts available in your system.

Default Job Server

If a repository is associated with several Job Servers, define one Job Server as the default Job Server to use at logon.

i Note

Job-specific options and path names that appear in the environment settings refer to the current default Job Server. If you change the default Job Server, modify these options and path names.

Option	Description
<i>Current</i>	Displays the current default Job Server.
<i>New</i>	Specifies a new default Job Server. Select a Job Server from the dropdown list. The software includes only the Job Servers that are associated with the local repository in the list. If you select a new Job Server, the software implements the change immediately.

Designer Communication Ports

Option	Description
<i>Allow Designer to set the port for Job Server communication</i>	Specifies whether Designer automatically sets an available port to receive messages from the current Job Server. The default setting is selected. Deselect the option to specify a listening port or port range. When you deselect this option, the software enables the options to set <i>From</i> and <i>To</i> port numbers.
<i>From</i> <i>To</i>	Specifies a range of ports. The software selects a port from the range to receive messages from the current Job Server. To specify a specific listening port, enter the same port number in both the <i>From</i> port and <i>To</i> port text boxes. To enable the changes, restart Designer. A good reason to set a range of ports instead of allowing Designer to set the port is when your system is set so that a firewall separates the Designer and the Job Server.
<i>Interactive Debugger</i>	Specifies a port for Debug mode. Designer uses the specified port to communicate with the Job Server while running in Debug mode.

Option	Description
<i>Server group for local repository</i>	Displays the name of the server group only when the repository is associated with a server group.

Related Information

[Changing the interactive debugger port \[page 836\]](#)

[General and environment options \[page 34\]](#)

4.7.2 Designer General options

Set options in the *General* group to define default settings for commonly used options in Designer.

Option	Description
<i>View data sampling size (rows)</i>	Specifies the sample size the software uses to display data from sources and target objects in open data flows in the workspace. View sample data in a source or target by clicking the magnifying glass icon in the lower left corner of the object in the data flow.
<i>Number of characters in workspace icon name</i>	Specifies the size for object name display in the workspace. The default setting is 17 characters. The software displays object names above each object in the workspace. The actual name can exceed the number of characters that you set here, but the software only displays the specified number of characters in the workspace, so the full name may not display.
<i>Maximum schema tree elements to auto expand</i>	Specifies the number of elements the software displays in the <i>Schema In</i> and <i>Schema Out</i> trees in data flow objects such as transforms. Enter a number for the <i>Input schema</i> and the <i>Output schema</i>. The default is 100. The software does not allow element names to exceed the set numbers.

Option	Description
<i>Default parameters to variables of the same name</i>	Specifies that the software automatically passes a variable as a parameter with the same name to a data flow called by a work flow. Applicable when you declare the variable at the work flow level.
<i>Automatically import domains</i>	Specifies that the software automatically imports the domain when it imports a table that references a domain.
<i>Perform complete validation before job execution</i>	Specifies that the software completely validates a job before it executes the job. The default setting is not checked. If you keep the default setting and do not select this option, ensure that you validate your job manually before you execute the job.
<i>Open monitor on job execution</i>	Specifies that the software switches the workspace to the monitor view during job execution. If not selected, the software maintains the workspace view during job execution. The default setting is checked. The job monitor displays each step of the job execution as the job executes.
<i>Automatically calculate column mappings</i>	Specifies that the software calculates information about target table column mapping based on the connected source or sources in the data flow. The software uses column-mapping information for meta-data reports such as impact and lineage, auto documentation, or custom reports. The software stores column-mapping information in the AL_COLMAP_NAMES table and the internal ALVW_MAPPING view after the following actions: • You save a data flow • You import objects to a repository • You export objects from a repository The software skips data flows that have errors. Therefore, if you select this option, ensure that you validate your entire job before you save it to prevent the software from skipping the data flows that have validation problems. Manually calculate column mappings in either Designer or the Management Console. For more information about manually calculating column mappings, see the Impact and Lineage section of the <i>Management Console Guide</i>.
<i>Show dialog when job is completed</i>	Specifies that the software displays a message when the job completes processing. If you do not select this option, the only way to know if the job has completed is by viewing the trace messages.

Option	Description
Show tabs in workspace	Specifies that the software displays object workspace tabs at the bottom of the workspace pane. Use tabs to identify what is open in the workspace, and to navigate between multiple open objects in the workspace.
Single window in workspace	Specifies that the software closes an open object in the workspace when you open a different object. Select this option to have only one object open at a time in the workspace.
Show Start Page at startup	Specifies whether the software displays the Start Page in the workspace pane when you log in to the software. The Start Page contains links to recent projects, links to documentation, and a link to open the Management Console. You can close the Start Page at any time. When you select this option, Designer reopens the Start Page each time you log in.
Enable Object Description when instantiate	Specifies that the software display object descriptions for objects in the workspace. Applicable only when an object has a description.
Exclude non-executable elements from export to XML Document	Specifies that the software excludes elements that do not affect job execution from exported XML documents. For example, when you select this option, the software does not export display coordinates in the workspace.

Related Information

[General and environment options \[page 34\]](#)

4.7.3 Designer Graphics options

Use settings in the [Graphics](#) options to choose and preview stylistic elements to customize your workspaces.

Option	Description
Workspace flow type	Specifies the flow type that your settings affect. When you create different settings for each flow type, you can visually determine the flow type open in the workspace. • Data Flow: Options affect the design of objects in data flows. • Job/Work Flow: Options affect the design of objects in job flows and work flows.

Option	Description
Line Type	Specifies the style for connecting lines between objects. • Straight: Lines are straight even when the line slants up or down. • Horizontal/Vertical: The software creates angles up or down when the line slants up or down. • Curved: The software curves the line when the line slants up or down.
Line Thickness	Specifies how thick the lines appear in the workspace. The software displays an example based on your choice of Thin, Medium, or Thick.
Background style	Specifies the background of the workspace pane. • Plain: Background appears without grid lines. • Tile: Background appears with grid lines with a white or gray background color. Background appears with evenly spaced dots with a blue background.
Color scheme	Specifies the color of the workspace pane. • Blue • Gray • White The software displays an example based on your choice.
Use navigation watermark	Specifies that the software adds a watermark graphic to the lower right of the workspace pane that indicates the type of flow. Available only for Plain Background style.

Related Information

[General and environment options \[page 34\]](#)

4.7.4 Designer Fonts options

Use the [Font](#) options to set font type, style, and size for text in the workspace pane.

For each option, click [Choose Font](#) to open the [Font](#) dialog box. Options include [Font](#), [Font style](#), and [Size](#).

Option	Description
Object label in workspace	Specifies the font options for the text that appears above each object in the workspace, such as the object name.
Object description and annotation in workspace	Specifies the font options for the text of the object description and annotations.
Custom controls	Specifies the font options for text other than object labels, descriptions, and annotations.

Related Information

[General and environment options \[page 34\]](#)

4.7.5 Designer Attribute Values options

Set *Attribute Values* options to preserve or clear existing attribute values for columns or tables when you reimport data.

First select the *Object Type*. Choose either *Column* or *Table*.

i Note

Some attributes are applicable to only columns or tables and not both object types.

Then select whether to preserve or clear the attribute value.

Attribute	Object type	Description
<i>Business_Description</i>	Column and Table	A business-level description of a table or column. The default setting is <i>Preserve</i> .
<i>Business_Name</i>	Column and Table	A logical field. This attribute defines and runs jobs that extract, transform, and load physical data while the Business Name data remains intact. The default setting is <i>Preserve</i> .
<i>Content_Type</i>	Column	Definition of the type of data in a column. The default setting is <i>Preserve</i> .
<i>Description</i>	Column and Table	Description of the column or table. The default setting is <i>Clear</i> .
<i>Estimated_Row_Count</i>	Table	An estimate of the table size that the software uses to calculate the order in which it reads tables to perform join operations. The default setting is <i>Preserve</i> .
<i>ObjectLabel</i>	Column and Table	A label that describes an object. The default setting is <i>Preserve</i> .
<i>Table_Usage</i>	Table	A label field that marks a table as fact or dimension, for example. The default setting is <i>Preserve</i> .

Related Information

[General and environment options \[page 34\]](#)

4.7.6 Designer Central Repository Connections options

Use the [Central Repository Connections](#) options to define default connection settings for the central repository.

The central repository provides a shared object library allowing developers to check objects in and out of their local repositories. Not all environments have a central repository. The administrator activates a central repository using the [Central Repository Connections](#) options.

Options	Description
Central Repository Connections	Displays the central repository connection. Click Add to add a different central repository connection. If the user does not have any other central repositories established, the software displays a message and opens a new logon dialog box to try another system or user. To activate the listed central repository, right-click the central repository and select Activate from the dropdown list.
Reactivate automatically	Specifies that the software reactivates the connection to the central repository when ever the user logs on to Designer using the current local repository.

Related Information

[Activate a central repository \[page 1003\]](#)

[General and environment options \[page 34\]](#)

4.7.7 Designer Language options

Set [Language](#) options to set a locale other than the default locale of English.

The software uses locale settings to display the Designer user interface and any text that the user interface generates in the set locale language.

Option	Description
Product locale	Specifies the language with which to display the Designer user interface and all product messages.
Preferred viewing locale	Specifies the language with which the software displays user data. For example, the software presents date formats in the preferred viewing locale.

Related Information

[General and environment options \[page 34\]](#)

4.7.8 Designer SSL options

Specify options in SSL to change the default SSL certificate and key file settings established when your administrator installed the software.

Only change the default settings to use your own SSL certificates and keyfiles.

i Note

If you change any SSL options other than [Use SSL protocol for profiler](#), Ensure that you restart both the Designer and any Data Services servers.

Option	Description
Server certificate file	Specifies the path to the server certificate file. The software requires that the server certificate is in PEM format.
Server private key file	Specifies the path to the server private key file.
Use server private key password file	Specifies to use a private key password file. When you select this option, also provide the location of the password file.
Trusted certificates folder	Specifies the location of the trusted certificates folder. Valid extensions for certificates in the trusted certificates folder include .pem, .crt, and .cer. The software requires the certificate file is in PEM format regardless of the folder type.
Use SSL protocol for profiler	Specifies that the software use SSL protocol for communications between the Designer and the profiler server.

Related Information

[General and environment options \[page 34\]](#)

4.7.9 Data General options

Use the options in [General](#) to set default data options in the software.

Option	Description										
Century Change Year	Specifies how the software interprets the century for two-digit years: - Two-digit years greater than or equal to this value are interpreted as 19##. - Two-digit years less than this value are interpreted as 20##. The default value is 15. ❖ Example If the Century Change Year is set to 15: <table><tr><th>Two-digit year</th><th>Interpreted as</th></tr><tr><td>99</td><td>1999</td></tr><tr><td>16</td><td>1916</td></tr><tr><td>15</td><td>1915</td></tr><tr><td>14</td><td>2014</td></tr></table>	Two-digit year	Interpreted as	99	1999	16	1916	15	1915	14	2014
Two-digit year	Interpreted as										
99	1999										
16	1916										
15	1915										
14	2014										
Convert blanks to nulls for Oracle bulk loader	Specifies that the software converts blanks to NULL values when you load data with the Oracle bulk loader utility under the following circumstances: - The column is not part of the primary key - The column is nullable										

Related Information

[General and environment options \[page 34\]](#)

4.7.10 Job Server Environment option

Set the [Environment](#) option to control the number of engine processes that SAP Data Services performs in parallel.

Data Services uses processes and threads to execute jobs that extract data from sources, transform the data, and loads data into a data warehouse. The number of concurrent processes and threads in a data flow affects job performance.

Option	Description
<i>Maximum number of engine processes</i>	Specifies the maximum number of engine processes that this Job Server can run concurrently. The default value is 2. This option is the global equivalent to the <i>Degree of parallelism</i> (DOP) option in the data flow <i>Properties</i> dialog box. For more information about working with both global and local DOP settings, see the <i>Performance Optimization Guide</i>.

Related Information

[General and environment options \[page 34\]](#)

[Degree of parallelism](#)

4.7.11 Job Server General options

Set *General* options to change default option values for an individual Job Server.

After you select a Job Server, change the default value for a select *Section* and *Key* combination.

❖ Example

To change the default setting of 0 for the number of times a Job Server tries to make an FTP connection if it initially fails, make the following settings:

Section: AL_ENGINE

Key: FTPNumberOfRetry

Value: 2

Enter a *Section* and a *Key*, and then enter an applicable value for *Value*.

The following table describes the applicable value pairs to choose.

Section	Key
int	AdapterDataExchangeTimeout
int	AdapterStartTimeout
AL_JobServer	AL_JobServerLoadBalanceDebug
AL_JobServer	AL_JobServerLoadOSPolling
string	DisplayDIInternalJobs
AL_Engine	FTPNumberOfRetry
AL_Engine	FTPRetryInterval

Section	Key
AL_Engine	Global_DOP
AL_Engine	IgnoreReducedMsgType
AL_Engine	IgnoreReducedMsgType_foo
AL_Engine	OCIServerAttach_Retry
AL_Engine	SPLITTER_OPTIMIZATION
AL_Engine	UseExplicitDatabaseLinks
Repository	UseDomainName

Related Information

[Changing option values for an individual Job Server \[page 770\]](#)

[General and environment options \[page 34\]](#)

4.7.12 Setting SAP environment options

Set *Environment* options in SAP Data Services for processes that are specific to SAP applications.

1. In Designer, select **Tools > Options**.
2. In the *Options* dialog box, expand *SAP* and select *Environment*.
3. Make settings as described in the following table.

Option	Description
<i>Maximum number of rows buffered before transfer to Data Services</i>	Specifies the maximum number of rows that the software holds in the SAP buffer before writing to the data transport file. The default value is 5000 rows. To manage the amount of memory used by generated ABAP programs, select the number of rows to write in a given batch. Value applies to all SAP application connections initiated by Data Services.

Option	Description
<i>Prefix for ABAP program names (up to 2 characters)</i>	Specifies the prefix to use to differentiate between Data Services-generated ABAP program names from other ABAP program names. All ABAP program names begin with a Y or a Z. Specify any two-character combination that begins with either a Y or a Z. The default is <i>ZW</i>.
<i>Convert SAP null to null</i>	Specifies whether the software converts NULL values from SAP sources in data flows to database-specific NULL. • Select: Software converts NULL values from SAP sources in data flows to database-specific NULL. • Deselected: Disables this behavior. Deselected is the default setting.
<i>Disable background SAP job status in Data Services log</i>	Specifies whether Data Services disables trace log entries when it queries the status of jobs submitted in the background. Applicable only when you enable <i>Execute in background (batch)</i>.

5 Logging in to the Designer

When you log in to Designer, you also log in to the local repository.

Ensure that you have permission to access the local repository and that you have the repository name and password. For details about creating a repository, see the *Administrator Guide*.

Before you can log in to Designer, ensure that your administrator added you as a user in the Central Management Console (CMC).

1. In the *SAP Data Services Repository Login* dialog box, enter your user credentials.

Option	Description
<i>System—host[:port]</i>	Specify the CMS server name and optionally the port number.
<i>User name</i>	Specify your user name that you use to log in to the CMS.
<i>Password</i>	Specify the password related to the value in <i>User name</i> .
<i>Authentication</i>	Specify the authentication type used by the CMS. Options include the following: ◦ <i>Enterprise</i> ◦ <i>LDAP</i> ◦ <i>SAP</i> ◦ <i>Windows AD</i>

2. Click *Log on*.
The software connects to the CMS using your credentials. Then the software populates a list of your repositories.
3. Select the applicable local repository and click *OK*.
The software displays the *Repository Password* dialog box. You can change this default behavior and set your repository so that the software automatically logs you on by setting access rights in the CMC. See the *Administrator Guide* for more information.
4. Enter the repository password and click *OK*.
The software opens the Designer in your default layout.

When you have access to more than one repository, you can switch to a different repository while you are logged on to Designer. Right-click a blank area of the object library and select *Switch Object Library* from the dropdown menu. The software displays the *SAP Data Services Repository Login* dialog box. Select a different repository and enter the applicable password.

5.1 Version restrictions

Ensure that your repository version is compatible with the current version of SAP Data Services Designer.

Ensure that your repository version is compatible with the same **major release** as the Designer and that your repository version is less than or equal to the version of the Designer.

As illustrated in the following example of Data Services versioning, the numbers in the third position from the left indicate the major release designation.

For example, in SAP Data Services version 14.2.10.1748, the numbers mean the following:

- 14.2 is the release version.
- 10 is the major release version, also known as the service pack.
- 1748 is the minor version, also known as the patch version.

During logon, the software alerts you if it finds a mismatch between your Designer version and your repository version. After you log in, you can view the software and repository versions by selecting ► [Help](#) ► [About Data Services](#) ►.

If you are logged on to an incompatible version of the repository, some Designer features may not work.

5.2 Resetting users

If more than one person attempts to log on to a single repository, SAP Data Services Designer may require that you reset or log off one of the users to continue.

Designer displays a [Reset Users](#) dialog box that lists users and the time they logged in to the repository. Options include the following:

- [Reset Users](#): Clears the users in the repository and sets you as the currently logged in user.
- [Continue](#): Continues to log you in to the system regardless of who else may be connected.
- [Exit](#): Terminates the logon attempt and closes the session.

i Note

Only select [Reset Users](#) or [Continue](#) when you know that you are the only user connected to the repository. Subsequent changes could corrupt the repository.

6 Reserved words

There are specific words that you should not use for naming objects because they have special meaning for Data Services.

Do not use the words in the following table as names for work flows, data flows, transforms, or other design elements that you create. Additionally, avoid using the reserved words as user names when you create a Data Services repository. The words are reserved with any combination of upper- and lower-case letters.

If you have to use reserved words, enter them with double quotation marks as shown in the following example:

```
"PRIMARY"
```

To help you remember reserved words, Data Services Designer displays the editor text areas that contain the reserved word in blue. The following table lists the reserved words.

Reserved words

_AL_DEFINE	INTEGER	BY
_AL_IFDEF	IS	CASE
_AL_METADATA_ELEMENT	JOB SERVER	CHAR
_AL_TRAN_FUNCTION	LOCAL	CONCAT
_MEMORY	LEFTOUTERJOIN	CREATE
_SAP_INNER_JOIN	LOAD	DATAFLOW
ABAP_PROGRAM	LOOKUP	DATE
ACTAGUICOMMENT	MOD	DECIMAL
ALL	NOT	DEFAULT
AL_NESTED_TABLE	NUMERIC	DISTINCT
AL_REAL_TIME_DATAFLOW	ON	DOMAIN
AL_REPO_FUNCTION	ORDER	EMBEDDED_DATAFLOW
AL_UNNEST	OUTPUT	END
AL_UNSPECIFIED_PARAMAND	PARALLEL	ERROR
AS	PLAN	ERROR_STEP
BEGIN	PSFT_TREE	FIRM_NOISE_WORD
BULK	READ	FOREIGN

CALL	REFERENCES	FUNCTION
CATCH	RETURNS	FUNC_CHAR
CHARACTER	SAP_TREE	FUNC_DS
CONVERT	SESSION	GLOBAL
CUSTOM	SYSTEM	IN
DATABASE	TABLE	INT
DASTORE	TRANSFORM	INTERVAL
DATETIME	TRY	KEY
DECLARE	VARCHAR	LIKE
DESC	VIEW	LONG
DISTINCT_KEY	WHERE	NULL
DOUBLE	_AL_ELSE	OR
ELSE	_AL_MESSAGE	OUT
EMBEDDED_DATAFLOW_RT	_AL_STORED_PROCEDURE	PIPE
END_TRY	_FUNC_TABLE	PRIMARY
ERROR_CONDITION	_RFC_FUNCTION	REAL
FILE	_SAP_LEFT_OUTER_JOIN	RETURN
FLOWOUTPUT	ACTA	ROW
FROM	ALGUICOMMENT	SELECT
FUNC_ANY	AL_NEST	SET
FUNC_COL	AL_PROJECT	SYSTEM_PROFILE
FUNC_NUM	AL_RELATION	TIME
GENERATED	AL_RFC_SCHEMA_GROUP	TRANSFORM_SCHEMA_MAPPING
GROUP	AL_UNNEST_SCHEMA_GROUP	VARIABLE
HAVING	AND	VOID
IF	ASC	WHILE
INPUT	BEGIN_SCRIPT	

7 Objects

All “entities” that you define, edit, or work with in SAP Data Services Designer are called objects.

Each object falls into one of two classes:

- Single use
- Reusable

The object class determines how you create and retrieve the object.

i Note

For information about source-specific objects, consult the applicable supplement document for that source. For example, for information about SAP applications as a source, consult the *Supplement for SAP*.

[Reusable objects \[page 53\]](#)

Data Services stores reusable object definitions in the repository so that you can reuse the object multiple times.

[Single use objects \[page 54\]](#)

Create single use objects in context with reusable objects.

[Save reusable objects \[page 55\]](#)

When you save a reusable object, the software stores the language that describes the object in the repository.

[Object metadata \[page 57\]](#)

When you create an object, you associate the object with a set of options, properties, and attributes.

[Object descriptions \[page 58\]](#)

Data Services operates using objects, and there are numerous objects that combine for Data Services processes.

[Object hierarchy \[page 62\]](#)

Object relationships are hierarchical.

[Object naming conventions \[page 65\]](#)

A consistent naming convention for SAP Data Services objects helps you easily identify objects listed in an object hierarchy.

[Object editors \[page 66\]](#)

SAP Data Services Designer provides object editors to create and configure objects.

[Creating a reusable object in the object library \[page 67\]](#)

Create a reusable object from the object library to use later in a project.

[Creating a reusable object using the tool palette \[page 68\]](#)

Create reusable and single-use objects from the tool palette while you have an object open in the workspace.

[Adding an existing object \[page 69\]](#)

Add an existing object to an open view in the workspace.

[Changing object names \[page 70\]](#)

Change the name of objects in the workspace, object library, or project area.

[Adding, changing, and viewing object properties \[page 71\]](#)

Add, view, and change the properties of an object through the *Properties* dialog box.

[Annotations \[page 74\]](#)

Use annotations like stick on notes to describe aspects of a flow of data such as in jobs, work flows, or data flows.

[Object descriptions \[page 75\]](#)

Include descriptions when you create objects to identify the purpose of the object.

[Cutting or copying objects \[page 78\]](#)

Cut or copy objects or calls to objects and paste them into a workspace where valid.

[Replicating objects \[page 79\]](#)

Replicating an object creates a copy of the object.

[Save and delete objects \[page 80\]](#)

When you save a reusable object in the software, you store the language that describes the object in the repository.

[Searching for objects \[page 83\]](#)

Search for objects that are defined in your repository or that are available through a datastore.

7.1 Reusable objects

Data Services stores reusable object definitions in the repository so that you can reuse the object multiple times.

A reusable object has a single definition. Reuse the definition of the object by creating calls to the definition. You access reusable objects through the object library.

If you change the definition of the object in one place, and then save the object, the software changes all other calls to the object.

❖ Example

You have a weekly load job and a daily load job that uses the same data flow. If you change the data flow in the weekly load job, the software changes the data flow in the daily load job as well.

When you drag and drop an object from the object library to your workspace, you create a new reference or call to the existing object definition. However, if you then edit the object using the object editor, and save the edits, the object also changes in your workspace.

❖ Example

You create a new project and add an existing data flow to the project. Then you open the data flow through the *Data Flow* tab in the object library and add an additional object. The instance of the data flow in your new project does not reflect the changes until you explicitly save the updated data flow.

Functions are reusable objects but they are not available in the object library. Access functions through the function wizard or smart editor wherever you can use the function.

i Note

Custom functions are available in the object library in the *Custom Functions* tab.

Some objects such as datastores and built-in transforms, are in the object library but are not reusable in all instances:

- Use Datastores in the object library to access the external metadata. You use the external metadata and not the datastore. The datastore is a method for categorizing and accessing external metadata.
- Built-in transforms are reusable in that every time you drop a transform in your workspace, the software creates a new instance of the transform.

Parent topic: [Objects \[page 52\]](#)

Related Information

[Single use objects \[page 54\]](#)

[Save reusable objects \[page 55\]](#)

[Object metadata \[page 57\]](#)

[Object descriptions \[page 58\]](#)

[Object hierarchy \[page 62\]](#)

[Object naming conventions \[page 65\]](#)

[Object editors \[page 66\]](#)

[Creating a reusable object in the object library \[page 67\]](#)

[Creating a reusable object using the tool palette \[page 68\]](#)

[Adding an existing object \[page 69\]](#)

[Changing object names \[page 70\]](#)

[Adding, changing, and viewing object properties \[page 71\]](#)

[Annotations \[page 74\]](#)

[Object descriptions \[page 75\]](#)

[Cutting or copying objects \[page 78\]](#)

[Replicating objects \[page 79\]](#)

[Save and delete objects \[page 80\]](#)

[Searching for objects \[page 83\]](#)

7.2 Single use objects

Create single use objects in context with reusable objects.

Access single use objects from the object in which they are created. Single use objects are not available from the object library. Single use objects, such as an annotation, or a Try Catch block, operate only in the context in which they are created.

The only way to save a single use object is by saving the object that contains the single use object. The software stores the language that describes the object to the repository.

❖ Example

Create an annotation while you have a data flow open. Select the annotation icon from the tool pallet and drop it onto the workspace. You save the annotation when you save the data flow.

The software saves the single use object description to the repository even if the object that contains the single use object does not validate.

Parent topic: [Objects \[page 52\]](#)

Related Information

[Reusable objects \[page 53\]](#)
[Save reusable objects \[page 55\]](#)
[Object metadata \[page 57\]](#)
[Object descriptions \[page 58\]](#)
[Object hierarchy \[page 62\]](#)
[Object naming conventions \[page 65\]](#)
[Object editors \[page 66\]](#)
[Creating a reusable object in the object library \[page 67\]](#)
[Creating a reusable object using the tool palette \[page 68\]](#)
[Adding an existing object \[page 69\]](#)
[Changing object names \[page 70\]](#)
[Adding, changing, and viewing object properties \[page 71\]](#)
[Annotations \[page 74\]](#)
[Object descriptions \[page 75\]](#)
[Cutting or copying objects \[page 78\]](#)
[Replicating objects \[page 79\]](#)
[Save and delete objects \[page 80\]](#)
[Searching for objects \[page 83\]](#)

7.3 Save reusable objects

When you save a reusable object, the software stores the language that describes the object in the repository.

The description of a reusable object includes these components:

- Properties of the object
- Options for the object

- Calls this object makes to other objects
- Definition of single-use objects called by this object

The transform stores the object description even if the software cannot successfully validate the object.

The software saves some reusable objects without prompting you to save them:

- When you import an object into the repository, the software saves the object.
- When you finish editing the following objects and close the editor, the software saves the object:
 - Datastores
 - Flat file formats
 - Nested schemas, such as DTD format, JSON format, or XML Schema

Save other reusable objects by using the [Save](#) or [Save All](#) options:

- Save the reusable object currently open in the workspace by choosing [Save](#) from the [Project](#) menu.

i Note

If a single-use object is open in the workspace, the [Save](#) command is not available.

- Save all objects in the repository that have changes by choosing [Save All](#) from the [Project](#) menu.

Additionally, the software prompts you to save all objects that have changes when you execute a job or when you exit Designer.

Parent topic: [Objects \[page 52\]](#)

Related Information

[Reusable objects \[page 53\]](#)
[Single use objects \[page 54\]](#)
[Object metadata \[page 57\]](#)
[Object descriptions \[page 58\]](#)
[Object hierarchy \[page 62\]](#)
[Object naming conventions \[page 65\]](#)
[Object editors \[page 66\]](#)
[Creating a reusable object in the object library \[page 67\]](#)
[Creating a reusable object using the tool palette \[page 68\]](#)
[Adding an existing object \[page 69\]](#)
[Changing object names \[page 70\]](#)
[Adding, changing, and viewing object properties \[page 71\]](#)
[Annotations \[page 74\]](#)
[Object descriptions \[page 75\]](#)
[Cutting or copying objects \[page 78\]](#)
[Replicating objects \[page 79\]](#)
[Save and delete objects \[page 80\]](#)
[Searching for objects \[page 83\]](#)

7.4 Object metadata

When you create an object, you associate the object with a set of options, properties, and attributes.

After you create and save an object, the software saves the options with metadata that identifies the object.

Object metadata

Object information	Description
Options	Defines the object operations such as database connection information, actions, passwords, and usernames. For example, in a datastore, an option is the name of the database to which the datastore connects. View and change object options by opening the object editor.
Properties	Documents object information such as object name and description. Properties describe an object, they do not affect the object operation. View object properties by right-clicking the object name and selecting Properties.
Attributes	Provides additional information about an object such as date created, method of saving, whether the object is enabled for Web service use. Attribute values may affect object behavior. View object attributes by opening the object properties and selecting the Attributes tab.

Parent topic: [Objects \[page 52\]](#)

Related Information

[Reusable objects \[page 53\]](#)
[Single use objects \[page 54\]](#)
[Save reusable objects \[page 55\]](#)
[Object descriptions \[page 58\]](#)
[Object hierarchy \[page 62\]](#)
[Object naming conventions \[page 65\]](#)
[Object editors \[page 66\]](#)
[Creating a reusable object in the object library \[page 67\]](#)
[Creating a reusable object using the tool palette \[page 68\]](#)
[Adding an existing object \[page 69\]](#)
[Changing object names \[page 70\]](#)
[Adding, changing, and viewing object properties \[page 71\]](#)
[Annotations \[page 74\]](#)
[Object descriptions \[page 75\]](#)
[Cutting or copying objects \[page 78\]](#)
[Replicating objects \[page 79\]](#)
[Save and delete objects \[page 80\]](#)

7.5 Object descriptions

Data Services operates using objects, and there are numerous objects that combine for Data Services processes.

The following table lists all objects in Data Services, the class of the object, and a description of the object. For more information about each object, see the individual object topics.

Object descriptions

Object	Object class	Description
Annotation	Single-use	Note that contains information about aspects of a flow, part of a flow, or a diagram in the workspace. You create the annotation and attach it to the item in the workspace.
Batch job	Reusable	Defines activities that the software executes at a given time, and includes error, monitor, and trace messages. Add jobs to projects only. After you save a batch job, the software places it in the object library. The batch job in the object library is a direct reference to the batch job object. Projects can contain references to multiple batch job objects, but only one reference per batch job.
Catch	Single-use	Specifies the steps to execute if an error occurs in a given exception group while a job is running.
COBOL copybook file format	Reusable	Defines the format for a COBOL copybook file source.
Conditional	Single-use	Specifies the steps to execute based on the result of a condition.
Data flow	Reusable	Specifies the requirements for extracting, transforming, and loading data from sources to targets. A data flow can be a part of a batch job or a real-time job.
Datastore	Reusable	Specifies the connection information to access a database or other data source. You cannot add the datastore object itself to the workspace.
Document	Reusable	Describes a data structure for complicated nested schemas. Available only for certain adapter datastore objects.
DTD	Reusable	Describes the format that an XML file or message reads or writes. DTD stands for Document Type Definition.
Excel workbook format	Reusable	Defines the format for an Excel workbook source.

Object	Object class	Description
File format	Reusable	Describes the location and file name of a flat file and the arrangement of flat file data in a source or target file.
File location	Reusable	Defines the file transfer protocol to use for transferring data files. Includes information about the remote and local servers. Associate a file location object to a file format to use as a source or target in a data flow. - As a source, the software uses the file location object information to transfer source data from a remote server to a local server. - As a target, the software uses the file location object information to transfer the output file from the local server to the remote server.
Function	Reusable	Returns a value.
HDFS file format	Reusable	Describes the structure of a Hadoop distributed file system.
JSON file	Single-use	A batch or real-time source or target. - As a source, a JSON file translates incoming JSON-formatted data into data that the software can process. - As a target, a JSON file translates the data produced by a data flow, including nested data, into a JSON-formatted file.
JSON message	Single-use	A real-time source or target. - As a source, JSON messages translate incoming JSON-formatted requests into data that a real-time job can process. - As a target, JSON messages translate the result of the real-time job, including hierarchical data, into a JSON-formatted response and sends the messages to the Access Server.
Log	Single-use	Records information about a particular execution of a single job.
Message function	Reusable	Accommodates XML messages when properly configured. Available only for certain adapter datastores.
Nested Schemas template	Single-use	A target that creates a JSON file or an XML file that matches a particular input schema. Does not require a DTD, JSON Schema, or XML Schema.
Outbound message	Reusable	XML-based, hierarchical communications that real-time jobs can publish to adapters. Available only for certain adapter datastores.
Project	Single-use	Contains groups of one or more jobs for convenient access.
Query transform	Single-use	Defines conditions to retrieve a specified data set.

Object	Object class	Description
Real time job	Reusable	Defines activities that the software executes on demand in real time. Create real time jobs in the Designer. Then configure and run as services associated with an Access Server in the Administrator. • Design real time jobs according to data flow model rules. • Run real time jobs as a request-response system.
Script	Single-use	Evaluates expressions, calls functions, and assigns values to variables.
Source	Single-use	Contains source data that the software reads and processes in a data flow.
Table	Reusable	Defines an external DBMS table for which metadata has been imported, or a target table into which data is or has been placed. A table is associated with a specific datastore. A table does not exist independently of a datastore connection. A table retrieves or stores data based on the schema of the table definition from which it was created.
Target	Single-use	Accepts extracted and transformed data in a data flow.
Template table	Reusable	Represents a new table that you add to a database. All datastores except SAP datastores have a default template that you use to create tables in the datastore. The software creates the schema for each instance of a template table at runtime. The software bases the created schema on the data loaded into the template table.
Transform	Reusable	Contains settings that perform specific operations on data sets. Requires zero or more data sets; produces zero or one data set, which can be split into more than one data set.
Try	Single-use	Introduces a Try/Catch block in a data flow. You can have more than one catch with a single try.
While loop	Single-use	Causes a sequence of steps to repeat as long as a set condition results in True.
Work flow	Reusable	Contains a specific order of data flows and operations that support a data flow.
XML file	Single-use	Translates XML data into applicable formats based on whether it is a source or target for batch and real time jobs. • As a source, translates incoming XML-formatted data into data that the software can process. • As a target, translates the data produced by a data flow, including nested data, into an XML-formatted file.

Object	Object class	Description
XML message	Single-use	Translates XML messages based on whether it is a source or target in real time jobs. - As a source, translates incoming XML-formatted requests into data that a real-time job can process. - As a target, translate the result of the real time job, including hierarchical data, into an XML-formatted response and sends the messages to the Access Server.
XML Schema	Reusable	Describes an XML file or message data structure so an XML document can read or write the data.

Parent topic: [Objects \[page 52\]](#)

Related Information

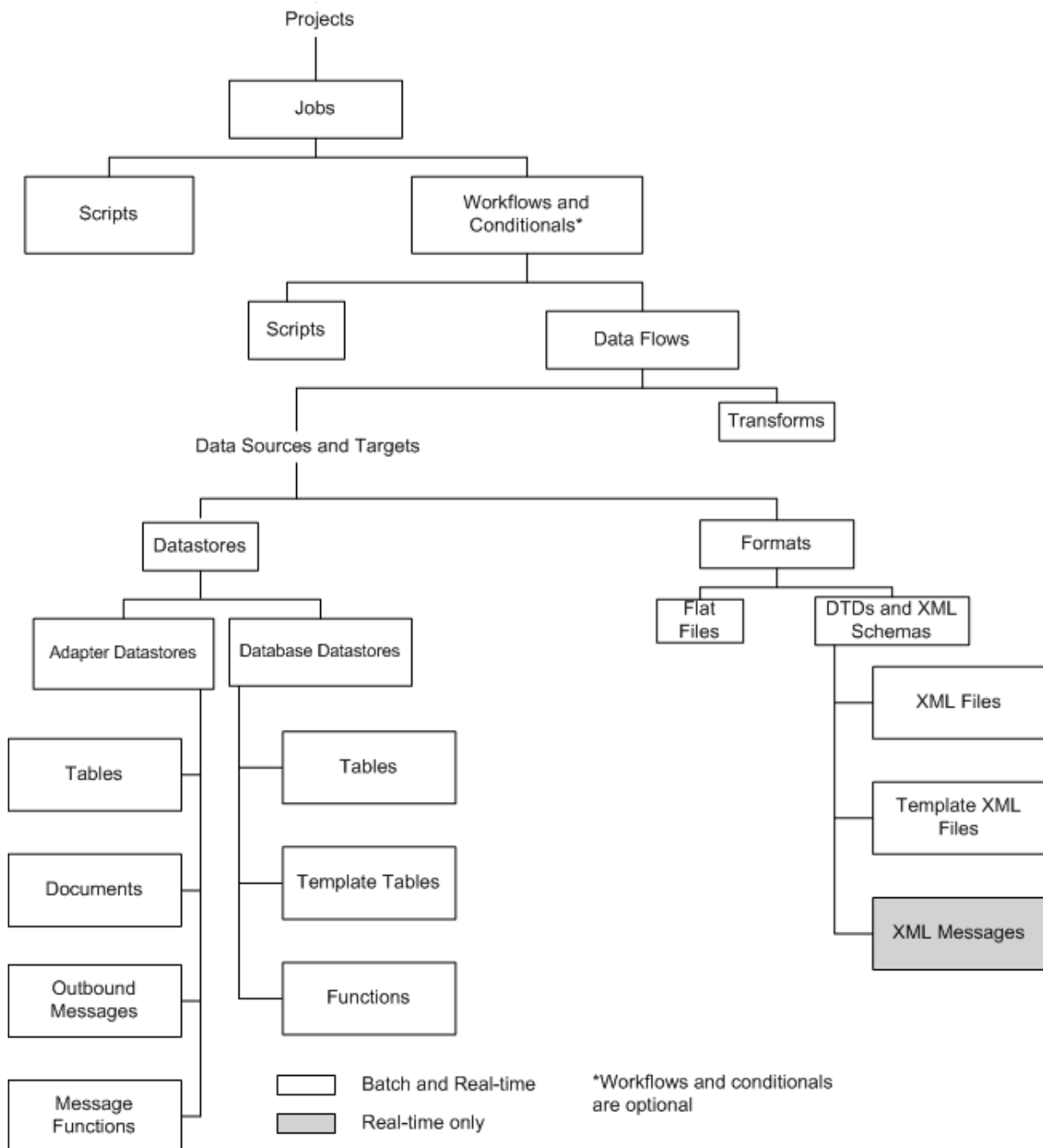
[Reusable objects \[page 53\]](#)
[Single use objects \[page 54\]](#)
[Save reusable objects \[page 55\]](#)
[Object metadata \[page 57\]](#)
[Object hierarchy \[page 62\]](#)
[Object naming conventions \[page 65\]](#)
[Object editors \[page 66\]](#)
[Creating a reusable object in the object library \[page 67\]](#)
[Creating a reusable object using the tool palette \[page 68\]](#)
[Adding an existing object \[page 69\]](#)
[Changing object names \[page 70\]](#)
[Adding, changing, and viewing object properties \[page 71\]](#)
[Annotations \[page 74\]](#)
[Object descriptions \[page 75\]](#)
[Cutting or copying objects \[page 78\]](#)
[Replicating objects \[page 79\]](#)
[Save and delete objects \[page 80\]](#)
[Searching for objects \[page 83\]](#)

7.6 Object hierarchy

Object relationships are hierarchical.

The highest object in the hierarchy is the project. The subordinate objects appear as nodes under a project. You add subordinate objects to the project in a specific order. For example, A project contains jobs, jobs contain workflows, and workflows contain data flows.

The following diagram shows the hierarchical relationships for the key object types within Data Services.



[Projects and subordinate objects \[page 63\]](#)

Projects contain jobs, work flows, and data flows as subordinate objects.

[Work flows \[page 64\]](#)

A work flow specifies the order in which SAP Data Services processes subordinate data flows.

[Data flows \[page 64\]](#)

A data flow is the process by which the software transforms source data into target data.

Parent topic: [Objects \[page 52\]](#)

Related Information

[Reusable objects \[page 53\]](#)

[Single use objects \[page 54\]](#)

[Save reusable objects \[page 55\]](#)

[Object metadata \[page 57\]](#)

[Object descriptions \[page 58\]](#)

[Object naming conventions \[page 65\]](#)

[Object editors \[page 66\]](#)

[Creating a reusable object in the object library \[page 67\]](#)

[Creating a reusable object using the tool palette \[page 68\]](#)

[Adding an existing object \[page 69\]](#)

[Changing object names \[page 70\]](#)

[Adding, changing, and viewing object properties \[page 71\]](#)

[Annotations \[page 74\]](#)

[Object descriptions \[page 75\]](#)

[Cutting or copying objects \[page 78\]](#)

[Replicating objects \[page 79\]](#)

[Save and delete objects \[page 80\]](#)

[Searching for objects \[page 83\]](#)

[About objects](#)

[Projects and subordinate objects](#)

[Data flows](#)

7.6.1 Projects and subordinate objects

Projects contain jobs, work flows, and data flows as subordinate objects.

A project is the highest-level object in Designer hierarchy. Projects provide a way to organize the subordinate objects, which are jobs, work flows, and data flows.

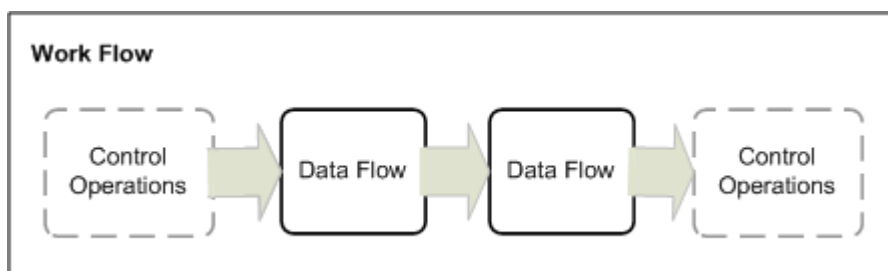
A project is open when you can view it in the project area. If you open a different project from the [Project](#) tab in the object library, the project area closes the current project and shows the project that you just opened.

Object	Subordinate objects	Subordinate description
Project	Job	The smallest unit of work that you can schedule independently for execution. Jobs are made up of work flows and data flows that direct the software in the order and manner of processing.
Job	Work flow	Incorporates data flows into a coherent flow of work for an entire job.
	Data flow	Process flow by which the software transforms source data into target data.

7.6.2 Work flows

A work flow specifies the order in which SAP Data Services processes subordinate data flows.

Arrange the subordinate data flows under the work flow so that the output from one data flow is ready for input to the intended data flow.



A work flow is a reusable object. It executes only within a Job. Use work flows to:

- Call data flows
- Call another work flow
- Define the order of steps to be executed in your job
- Pass parameters to and from data flows
- Define conditions for executing sections of the project
- Specify how to handle errors that occur during execution

Work flows are optional.

7.6.3 Data flows

A data flow is the process by which the software transforms source data into target data.

Data flows process data in the order in which they are arranged in the workspace.

A data flow defines the basic task that Data Services accomplishes. The basic task is moving data from one or more sources to one or more target tables or files.

You define data flows by identifying the sources from which to extract data, the transformations that the data should undergo, and the targets.



Use data flows to:

- Identify the source data to read
- Define the transformations to perform on the data
- Identify the target table to load data

A data flow is a reusable object. It is always called from a work flow or a job.

7.7 Object naming conventions

A consistent naming convention for SAP Data Services objects helps you easily identify objects listed in an object hierarchy.

The following table shows a suggested naming convention for your objects. We use this naming convention throughout the tutorial.

Object	Prefix	Suffix	Example
Job	JOB		JOB_SalesOrg
Work flow	WF		WF_SalesOrg
Data flow	DF		DF_Currency
Datastore		DS	ODS_DS

For more information about naming conventions for objects in jobs, see the Projects and Jobs section of the *Designer Guide*.

Parent topic: [Objects \[page 52\]](#)

Related Information

[Reusable objects \[page 53\]](#)

[Single use objects \[page 54\]](#)
[Save reusable objects \[page 55\]](#)
[Object metadata \[page 57\]](#)
[Object descriptions \[page 58\]](#)
[Object hierarchy \[page 62\]](#)
[Object editors \[page 66\]](#)
[Creating a reusable object in the object library \[page 67\]](#)
[Creating a reusable object using the tool palette \[page 68\]](#)
[Adding an existing object \[page 69\]](#)
[Changing object names \[page 70\]](#)
[Adding, changing, and viewing object properties \[page 71\]](#)
[Annotations \[page 74\]](#)
[Object descriptions \[page 75\]](#)
[Cutting or copying objects \[page 78\]](#)
[Replicating objects \[page 79\]](#)
[Save and delete objects \[page 80\]](#)
[Searching for objects \[page 83\]](#)

7.8 Object editors

SAP Data Services Designer provides object editors to create and configure objects.

An object editor provides all options and settings for you to create the object. Based on the object type, an editor can provide options to set up input and output schemas, processing options, connection options, and so on.

After you add an object to a data flow or work flow, open it in the workspace. Configure the object to play a specific role in the data flow or work flow.

For example, a data flow in the workspace may contain a source and target plus transforms or queries. Open each object editor to set options applicable to the specific data flow:

- A source editor may contain an option to enter a related datastore so that the data flow can access source data.
- A Query transform editor may contain settings to query the source data so that the data flow processes only specific types of records.
- A transform editor may contain settings that add additional information to your data.
- A target editor may contain settings to upload generated data to a specific location using a file location object for transfer protocol information.

The Reference Guide contains descriptions for all options in object editors. When you create or edit an object, consult the *Reference Guide* for descriptions of options.

Parent topic: [Objects \[page 52\]](#)

Related Information

[Reusable objects \[page 53\]](#)
[Single use objects \[page 54\]](#)
[Save reusable objects \[page 55\]](#)
[Object metadata \[page 57\]](#)
[Object descriptions \[page 58\]](#)
[Object hierarchy \[page 62\]](#)
[Object naming conventions \[page 65\]](#)
[Creating a reusable object in the object library \[page 67\]](#)
[Creating a reusable object using the tool palette \[page 68\]](#)
[Adding an existing object \[page 69\]](#)
[Changing object names \[page 70\]](#)
[Adding, changing, and viewing object properties \[page 71\]](#)
[Annotations \[page 74\]](#)
[Object descriptions \[page 75\]](#)
[Cutting or copying objects \[page 78\]](#)
[Replicating objects \[page 79\]](#)
[Save and delete objects \[page 80\]](#)
[Searching for objects \[page 83\]](#)

7.9 Creating a reusable object in the object library

Create a reusable object from the object library to use later in a project.

1. Open the applicable object tab in the object library.
2. Right-click an empty area of the open tab and select [New](#) from the dropdown menu.
3. Based on the object type, you may need to make another selection. For example, when you create a new data flow, you select to create either a data flow or an ABAP data flow.

The object appears in the list of objects in the object library tab.

4. Right-click the new object in the object library and select [Properties](#).

The [Properties](#) dialog box opens.

5. Enter an object name and description to identify the object.

Based on the type of object, there are other options to set. For example, when you create a new data flow, you set whether to execute the data flow only once, the degree of parallelism, and the cache type.

6. Click [OK](#) to close [Properties](#).

Task overview: [Objects \[page 52\]](#)

Related Information

[Reusable objects \[page 53\]](#)
[Single use objects \[page 54\]](#)
[Save reusable objects \[page 55\]](#)
[Object metadata \[page 57\]](#)
[Object descriptions \[page 58\]](#)
[Object hierarchy \[page 62\]](#)
[Object naming conventions \[page 65\]](#)
[Object editors \[page 66\]](#)
[Creating a reusable object using the tool palette \[page 68\]](#)
[Adding an existing object \[page 69\]](#)
[Changing object names \[page 70\]](#)
[Adding, changing, and viewing object properties \[page 71\]](#)
[Annotations \[page 74\]](#)
[Object descriptions \[page 75\]](#)
[Cutting or copying objects \[page 78\]](#)
[Replicating objects \[page 79\]](#)
[Save and delete objects \[page 80\]](#)
[Searching for objects \[page 83\]](#)

7.10 Creating a reusable object using the tool palette

Create reusable and single-use objects from the tool palette while you have an object open in the workspace.

1. Click the object in the tool palette and click in a blank area of the workspace.

The action is not a drag and drop motion. When you click the object in the tool palette, your mouse pointer changes when you hover over the workspace.

The software creates an object in the workspace. The software automatically selects the existing temporary title so that you can add a new name.

2. Enter a new name for the object.

The software adds the object to the project area under the object to which you added the new object. The software also adds the object to the object library.

If you delete the object from the workspace, the software removes it from the project area, but the object remains available in the library.

Task overview: [Objects \[page 52\]](#)

Related Information

[Reusable objects \[page 53\]](#)
[Single use objects \[page 54\]](#)
[Save reusable objects \[page 55\]](#)
[Object metadata \[page 57\]](#)
[Object descriptions \[page 58\]](#)
[Object hierarchy \[page 62\]](#)
[Object naming conventions \[page 65\]](#)
[Object editors \[page 66\]](#)
[Creating a reusable object in the object library \[page 67\]](#)
[Adding an existing object \[page 69\]](#)
[Changing object names \[page 70\]](#)
[Adding, changing, and viewing object properties \[page 71\]](#)
[Annotations \[page 74\]](#)
[Object descriptions \[page 75\]](#)
[Cutting or copying objects \[page 78\]](#)
[Replicating objects \[page 79\]](#)
[Save and delete objects \[page 80\]](#)
[Searching for objects \[page 83\]](#)

7.11 Adding an existing object

Add an existing object to an open view in the workspace.

Open an object such as a data flow in your workspace.

1. Open the applicable tab in the object library.
2. Select the applicable object.

i Note

Ensure that the object is lower in hierarchy order than the object open in the workspace.

3. Drag and drop the object to the workspace.

i Note

The software displays a warning icon and does not move the object when you try to add an object to an object view that is of a higher hierarchy.

Configure the object that you added. For example, if you add a data flow, add additional objects to the data flow, connect the objects, and save the data flow.

Task overview: [Objects \[page 52\]](#)

Related Information

[Reusable objects \[page 53\]](#)
[Single use objects \[page 54\]](#)
[Save reusable objects \[page 55\]](#)
[Object metadata \[page 57\]](#)
[Object descriptions \[page 58\]](#)
[Object hierarchy \[page 62\]](#)
[Object naming conventions \[page 65\]](#)
[Object editors \[page 66\]](#)
[Creating a reusable object in the object library \[page 67\]](#)
[Creating a reusable object using the tool palette \[page 68\]](#)
[Changing object names \[page 70\]](#)
[Adding, changing, and viewing object properties \[page 71\]](#)
[Annotations \[page 74\]](#)
[Object descriptions \[page 75\]](#)
[Cutting or copying objects \[page 78\]](#)
[Replicating objects \[page 79\]](#)
[Save and delete objects \[page 80\]](#)
[Searching for objects \[page 83\]](#)

7.12 Changing object names

Change the name of objects in the workspace, object library, or project area.

i Note

You cannot change the names of built-in objects.

The following steps show two ways to change an object name: Using *Rename* or *Properties*.

1. To change the name of an object in the object library, workspace, or project area using *Rename*:
 - a. Right-click the object and select *Rename* from the dropdown menu.

The current name of the object appears in a text box.
 - b. Edit the text in the name text box.
 - c. Click outside the text box or press to save the new name.
2. To change the name of an object in the object library, workspace, or project area using *Properties*:
 - a. Right-click the object and select *Properties* from the dropdown menu.

The *Properties* dialog box opens.
 - b. Enter the new name in the *Name* text box.
 - c. Optional. Enter a description in the *Description* text box.
 - d. Complete other options as applicable.

- e. Click *OK*.

The object name change affects all instances of the object. If the object is included in two projects, the software changes the name of the object in both projects. The software also changes the name of the object in the object library.

Task overview: [Objects \[page 52\]](#)

Related Information

[Reusable objects \[page 53\]](#)
[Single use objects \[page 54\]](#)
[Save reusable objects \[page 55\]](#)
[Object metadata \[page 57\]](#)
[Object descriptions \[page 58\]](#)
[Object hierarchy \[page 62\]](#)
[Object naming conventions \[page 65\]](#)
[Object editors \[page 66\]](#)
[Creating a reusable object in the object library \[page 67\]](#)
[Creating a reusable object using the tool palette \[page 68\]](#)
[Adding an existing object \[page 69\]](#)
[Adding, changing, and viewing object properties \[page 71\]](#)
[Annotations \[page 74\]](#)
[Object descriptions \[page 75\]](#)
[Cutting or copying objects \[page 78\]](#)
[Replicating objects \[page 79\]](#)
[Save and delete objects \[page 80\]](#)
[Searching for objects \[page 83\]](#)

7.13 Adding, changing, and viewing object properties

Add, view, and change the properties of an object through the *Properties* dialog box.

1. Right-click the object in the object library and select *Properties* from the dropdown menu.

The *General* tab of the *Properties* dialog box opens.

2. Complete each tab as applicable.

The property tabs vary by object type. The most common tabs are *General*, *Attributes*, and *Class Attributes*.

3. Click *OK* to save your settings and to close the *Properties* dialog box.

Alternatively, click *Apply* to save changes without closing the dialog box.

[Object Properties, General tab \[page 72\]](#)

Use the *General* tab to set the object name, description, and other settings based on the object type.

[Object Properties, Attributes tab \[page 73\]](#)

In the *Attributes* tab, view and edit object attributes.

[Object Properties, Class Attributes tab \[page 73\]](#)

The *Class Attributes* tab displays the class attributes for the type of object selected.

Task overview: [Objects \[page 52\]](#)

Related Information

[Reusable objects \[page 53\]](#)

[Single use objects \[page 54\]](#)

[Save reusable objects \[page 55\]](#)

[Object metadata \[page 57\]](#)

[Object descriptions \[page 58\]](#)

[Object hierarchy \[page 62\]](#)

[Object naming conventions \[page 65\]](#)

[Object editors \[page 66\]](#)

[Creating a reusable object in the object library \[page 67\]](#)

[Creating a reusable object using the tool palette \[page 68\]](#)

[Adding an existing object \[page 69\]](#)

[Changing object names \[page 70\]](#)

[Annotations \[page 74\]](#)

[Object descriptions \[page 75\]](#)

[Cutting or copying objects \[page 78\]](#)

[Replicating objects \[page 79\]](#)

[Save and delete objects \[page 80\]](#)

[Searching for objects \[page 83\]](#)

7.13.1 Object Properties, General tab

Use the *General* tab to set the object name, description, and other settings based on the object type.

The *General* tab of the *Properties* dialog box contains two main properties: *Name* and *Description*.

Depending on the object type, there may be other options to set in the *General* tab:

- *Execute only once*
- *Recover as a unit*
- *Degree of parallelism*
- *Use database links*
- *Cache type*

- [Bypass](#)

For complete descriptions of specific object property settings, see the *Reference Guide*.

7.13.2 Object Properties, Attributes tab

In the [Attributes](#) tab, view and edit object attributes.

To assign a value to an attribute, select the attribute and enter the value in the [Value](#) box at the bottom of the window.

Some attribute values are set by the software and cannot be edited. When you select an attribute with a system-defined value, the [Value](#) field is unavailable. Some attributes include the following:

- ANSI Varchar DQ
- Column Mapping Calculated
- Date created
- Date modified
- Saved after check out

There are different attributes based on the object type. For more information about attributes for a specific object, see the *Reference Guide*.

7.13.3 Object Properties, Class Attributes tab

The [Class Attributes](#) tab displays the class attributes for the type of object selected.

All objects of the same type contain the same class attributes. All datastores have the same class attributes, for example.

To create a new attribute for a class of objects, right-click in the attribute list and select [Add](#). The new attribute is now available for all of the objects of this class.

To delete an attribute, select it then right-click and choose [Delete](#).

i Note

You cannot delete the class attributes predefined by Data Services.

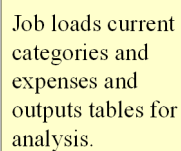
7.14 Annotations

Use annotations like stick on notes to describe aspects of a flow of data such as in jobs, work flows, or data flows.

Together, object descriptions and annotations allow you to document an SAP Data Services application. For example, you can describe the incremental behavior of individual jobs with annotations and include a description for each object.

❖ Example

For example, the following annotation describes the purpose of a job:



Job loads current categories and expenses and outputs tables for analysis.

Annotations are different than object descriptions because annotations are related to a process or application and descriptions are related to individual objects.

When you import or export a job, work flow, or data flow, you import or export associated annotations.

Parent topic: [Objects \[page 52\]](#)

Related Information

[Reusable objects \[page 53\]](#)
[Single use objects \[page 54\]](#)
[Save reusable objects \[page 55\]](#)
[Object metadata \[page 57\]](#)
[Object descriptions \[page 58\]](#)
[Object hierarchy \[page 62\]](#)
[Object naming conventions \[page 65\]](#)
[Object editors \[page 66\]](#)
[Creating a reusable object in the object library \[page 67\]](#)
[Creating a reusable object using the tool palette \[page 68\]](#)
[Adding an existing object \[page 69\]](#)
[Changing object names \[page 70\]](#)
[Adding, changing, and viewing object properties \[page 71\]](#)
[Object descriptions \[page 75\]](#)
[Cutting or copying objects \[page 78\]](#)

[Replicating objects \[page 79\]](#)

[Save and delete objects \[page 80\]](#)

[Searching for objects \[page 83\]](#)

7.14.1 Creating and deleting annotations

Create or delete an annotation in the workspace in SAP Data Services Designer.

Ensure that viewing annotations is enabled in your workspace by selecting  [View](#) .

1. Open a process such as a job or data flow in your workspace.
2. Click the [Annotation](#) icon in the tool palette and click an empty area of the workspace.

A yellow text box appears.

3. Enter the applicable text.

Resize the text box by clicking the box and dragging the edges to the desired size.

4. Drag and drop the note on the applicable object or portion of the data flow.
5. To delete the annotation, right-click the annotation and select [Delete](#) from the dropdown menu.

Related Information

[Object descriptions \[page 75\]](#)

7.15 Object descriptions

Include descriptions when you create objects to identify the purpose of the object.

Descriptions provide a convenient way to include comments to workspace objects. Descriptions also help you remember why you created an object, and how to use the object.

The software associates the object description with the object so that the description stays with the object. When you move an object in the workspace, the description also moves. When you import or export the repository that contains the object, you import or export the object description as well.

Annotations are different than object descriptions. Annotations describe aspects of a work flow or data flow. They are related to the process and not individual objects within the process.

Designer displays descriptions when you activate both system level and object level settings.

The system setting is unique to your setup, but it is disabled by default. The object setting is also disabled by default until you add or edit a description to the object from the workspace. The software saves the object-level setting with the object in the repository.

An ellipsis that appears after the text in a description indicates that there is more text to see. Resize the text box to see all of the text. You can also select an object description and view a complete description in the status bar. Find the status bar in the lower left of the *Designer* window.

[Adding a description to an object \[page 77\]](#)

Even though including a description for an object is optional, a description is an easy way to include descriptions in your workspace when you include the object in a data flow or process.

[Displaying a description in the workspace \[page 77\]](#)

View object descriptions in the workspace by setting a system and an object level option.

[Hiding object descriptions in the workspace \[page 77\]](#)

[Editing object descriptions in the workspace \[page 77\]](#)

Add, cut, copy, and paste changes in an existing object description.

Parent topic: [Objects \[page 52\]](#)

Related Information

[Reusable objects \[page 53\]](#)

[Single use objects \[page 54\]](#)

[Save reusable objects \[page 55\]](#)

[Object metadata \[page 57\]](#)

[Object descriptions \[page 58\]](#)

[Object hierarchy \[page 62\]](#)

[Object naming conventions \[page 65\]](#)

[Object editors \[page 66\]](#)

[Creating a reusable object in the object library \[page 67\]](#)

[Creating a reusable object using the tool palette \[page 68\]](#)

[Adding an existing object \[page 69\]](#)

[Changing object names \[page 70\]](#)

[Adding, changing, and viewing object properties \[page 71\]](#)

[Annotations \[page 74\]](#)

[Cutting or copying objects \[page 78\]](#)

[Replicating objects \[page 79\]](#)

[Save and delete objects \[page 80\]](#)

[Searching for objects \[page 83\]](#)

[Annotations \[page 74\]](#)

7.15.1 Adding a description to an object

Even though including a description for an object is optional, a description is an easy way to include descriptions in your workspace when you include the object in a data flow or process.

1. In the project area, object library, or workspace, right-click an object and select *Properties*.
2. Enter comments in the *Description* text box.
3. Click *OK*.

The description for the object displays in the Description column in the object library.

7.15.2 Displaying a description in the workspace

View object descriptions in the workspace by setting a system and an object level option.

1. Open an existing object in the workspace. For example, open a job or a data flow. Ensure that there is an object in the opened workspace view that contains a description.
2. Select **View > Enabled Descriptions**.

Enabled Descriptions is a toggle option. It is enabled when it has a checkmark next to it in the menu. If it has a checkmark next to it already, do not select it again. You might disable it instead of enable it.

3. Right-click the specific object in the workspace and select *Enable Object Description* from the dropdown menu.

The description displays in the workspace under the object.

7.15.3 Hiding object descriptions in the workspace

1. Open an object in the workspace such as a job or data flow.
2. Right-click an object that shows a description. Select multiple objects by pressing **Ctrl** + **Shift**.
3. Right-click and deselect *Enable Object Description* in the dropdown menu.

The software hides the description for the selected object even when the *View Enabled Descriptions* option is enabled. The *View Enabled Descriptions* is the system-level setting. The object-level setting overrides the system-level setting.

7.15.4 Editing object descriptions in the workspace

Add, cut, copy, and paste changes in an existing object description.

1. In the workspace, double-click an object description.
2. Add, cut, copy, or paste text into the description.
3. Select **Project > Save**.

i Note

If you edit the description of a reusable object, the software alerts you that your change affects all occurrences of the object across all jobs. Before you select the *Don't show this warning next time* checkbox, consider that you can reactivate the warning only by calling Technical Support.

7.16 Cutting or copying objects

Cut or copy objects or calls to objects and paste them into a workspace where valid.

1. In the workspace, select the object or objects you want to cut or copy.

Select multiple objects by pressing `Ctrl` + `Shift`.

2. Right-click and select either *Cut* or *Copy* from the dropdown menu.

You can also press `Ctrl` + `C` to copy or `Ctrl` + `X` to cut.

3. Click within the same object view in the workspace or open a different object view in the workspace.
4. Right-click and select *Paste*.

When an object in the target location contains the same name, the software automatically generates a new name for the pasted object by adding a number at the end of the file name. For example, the software generates the following new name for a duplicate of the transform named MyTransform: MyTransform_1.

The software adds the objects to the new object or location in the selected location.

If you choose one of the following methods to paste the objects, the software places the objects in the upper left corner of the workspace:

- Click the *Paste* icon.
- Click **Edit > Paste**.
- Press `Ctrl` + `V`.

If you use the alternate methods to paste more objects into the new location, the software pastes the objects on top of the previous pasted objects. Ensure that you move the pasted objects before pasting more objects to avoid this behavior.

When you copy objects that contain references to global variables, local variables, parameters, and substitution parameters, ensure that you redefine them within each new context.

Task overview: [Objects \[page 52\]](#)

Related Information

[Reusable objects \[page 53\]](#)

[Single use objects \[page 54\]](#)

[Save reusable objects \[page 55\]](#)

[Object metadata \[page 57\]](#)

[Object descriptions \[page 58\]](#)

[Object hierarchy \[page 62\]](#)
[Object naming conventions \[page 65\]](#)
[Object editors \[page 66\]](#)
[Creating a reusable object in the object library \[page 67\]](#)
[Creating a reusable object using the tool palette \[page 68\]](#)
[Adding an existing object \[page 69\]](#)
[Changing object names \[page 70\]](#)
[Adding, changing, and viewing object properties \[page 71\]](#)
[Annotations \[page 74\]](#)
[Object descriptions \[page 75\]](#)
[Replicating objects \[page 79\]](#)
[Save and delete objects \[page 80\]](#)
[Searching for objects \[page 83\]](#)

7.17 Replicating objects

Replicating an object creates a copy of the object.

Replicate objects from the object library. The software generates a different name for the copy. The copy is separate object from the original. You can edit the copy without affecting the original object or any occurrences of the original object.

1. Select the applicable tab in the object library.
2. Right-click the object to be replicated.
3. Select *Replicate* from the dropdown menu.

The software copies the object and adds it to the list of objects. The software prefixes the original object name with `copy_x_transform_name`. For example, the first replicated copy is `copy_1_transform_name`, the second is `copy_2_transform_name`, and so on.

Task overview: [Objects \[page 52\]](#)

Related Information

[Reusable objects \[page 53\]](#)
[Single use objects \[page 54\]](#)
[Save reusable objects \[page 55\]](#)
[Object metadata \[page 57\]](#)
[Object descriptions \[page 58\]](#)
[Object hierarchy \[page 62\]](#)
[Object naming conventions \[page 65\]](#)
[Object editors \[page 66\]](#)

[Creating a reusable object in the object library \[page 67\]](#)
[Creating a reusable object using the tool palette \[page 68\]](#)
[Adding an existing object \[page 69\]](#)
[Changing object names \[page 70\]](#)
[Adding, changing, and viewing object properties \[page 71\]](#)
[Annotations \[page 74\]](#)
[Object descriptions \[page 75\]](#)
[Cutting or copying objects \[page 78\]](#)
[Save and delete objects \[page 80\]](#)
[Searching for objects \[page 83\]](#)

7.18 Save and delete objects

When you save a reusable object in the software, you store the language that describes the object in the repository.

Save single use objects as part of the definition of the reusable object that calls the single use object.

When you choose to save a reusable object in the workspace, the software saves the following information to the repository:

- Object properties
- Definitions for all single use objects that the object calls
- Calls to other reusable objects recorded in the repository

The software stores the description of the object in the repository even when the object is not complete or contains an error and does not validate.

Parent topic: [Objects \[page 52\]](#)

Related Information

[Reusable objects \[page 53\]](#)
[Single use objects \[page 54\]](#)
[Save reusable objects \[page 55\]](#)
[Object metadata \[page 57\]](#)
[Object descriptions \[page 58\]](#)
[Object hierarchy \[page 62\]](#)
[Object naming conventions \[page 65\]](#)
[Object editors \[page 66\]](#)
[Creating a reusable object in the object library \[page 67\]](#)
[Creating a reusable object using the tool palette \[page 68\]](#)
[Adding an existing object \[page 69\]](#)

[Changing object names \[page 70\]](#)
[Adding, changing, and viewing object properties \[page 71\]](#)
[Annotations \[page 74\]](#)
[Object descriptions \[page 75\]](#)
[Cutting or copying objects \[page 78\]](#)
[Replicating objects \[page 79\]](#)
[Searching for objects \[page 83\]](#)

7.18.1 Saving changes to single reusable objects

When you edit and save a reusable object, the software saves your changes to the object and all objects that the object calls.

Open the applicable project that contains the object to be edited. Open the object so it appears in the workspace. Make changes to the object in your workspace and perform the following steps to save your changes:

Select  **Project** > **Save** .

The software saves the changed information in the project, including changes that you made to the reusable objects.

7.18.2 Saving all changed objects in the repository

To ensure that all objects that you have changed in a session are saved to the repository, use the [Save All](#) option.

1. Choose  **Project** > **Save All** .

The software lists the reusable objects that were changed since the last save operation.

2. If applicable, deselect any listed object that you do not want saved. For example, maybe you were experimenting with a data flow and you decide not to save it.
3. Click [OK](#).

If you do not remember to select [Save All](#) to save changes since the last save, and you attempt to exit Designer, the software reminds you to save changes by presenting a list of all changed objects. The software does the same thing when you attempt to execute a job without first saving all of your changes.

7.18.3 About deleting objects

To delete an object, first decide whether to delete the object from the project or delete the object from the repository.

When you delete an object from a project in the project area, the software removes the object from the project. The object is still available in the object library and the repository.

When you delete the object from the object library, the software deletes all occurrences of the object from the repository. If the object is called in separate data flows, the software deletes the object from each data flow. The deletion may adversely affect all related objects.

To protect you from deleting objects unintentionally, the software issues a notice before it deletes the object from the repository. The notice states that the object is used in multiple locations, and it provides the following options:

- [Yes](#): Continues with the delete of the object from the repository.
- [No](#): Discontinues the delete process.
- [View Where Used](#): Displays a list of the related objects in which the object will be deleted.

7.18.4 Deleting an object definition from the repository

To delete an object from the repository, delete the object from the Designer object library.

i Note

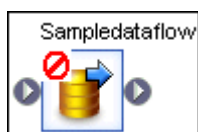
The software does not allow you to delete built-in objects such as transforms from the object library.

1. In the object library, find the object to be deleted.
2. Right-click and select [Delete](#) from the dropdown menu.

When the object is used by other objects, the software issues a warning and provides options to continue with the deletion, cancel the deletion, or to view the list of objects that currently use the object. When the object is not related to any other object, a confirmation message appears.

3. For related objects, select either [Yes](#), [No](#), or [View where used](#).

If you select [Yes](#), the software marks all calls to the object with an icon indicating the object was deleted and the call is invalid.



If you select [View where used](#), the software lists all objects that call the object to be deleted. Close the list by clicking the [X](#) icon in the upper corner of the message. Make replacements in the affected objects so that deleting the object does not adversely affect the object. Try to delete the object again.

4. For single, unrelated objects, click [Yes](#) to close the confirmation message and delete the object.

The software deletes the object information from the repository.

If you did not replace the object to be deleted before you delete the object, make sure that you update the calls after the delete process completes.

Related Information

[Using View Where Used \[page 815\]](#)

7.18.5 Deleting an object call

Delete an object call from the workspace so the object is no longer related to the calling object.

1. Open the object that contains the call you want to delete.
2. Right-click the object and choose [Delete](#).

If you delete a reusable object from the workspace or from the project area, only the object call is deleted. The object definition remains in the object library.

7.19 Searching for objects

Search for objects that are defined in your repository or that are available through a datastore.

1. Right-click in any tab of the object library and choose [Search](#).
The [Search](#) dialog box opens.
2. Enter the appropriate values for the search based on descriptions in the following table.

Option	Description
Look in	Specifies where to search for the object. The dropdown list includes all existing datastores and your repository. When you select a datastore, the software enables the Internal Data and the External Data dropdown list.
Internal Data External Data	Specifies where to search in a datastore: ◦ Internal Data: Searches the imported data in the datastore. ◦ External Data: Searches the entire datastore.
Local Repository Central Repository	Specifies whether the software searches in the local repository or the central repository for the search item. If you do not have a central repository, the software disables the Central Repository option.

Option	Description
Object type	Specifies the type of object to find. The dropdown list displays objects related to what you choose for Look in. - When you search in the repository, select from a list of object types present in the repository. - When you search in a datastore, select from a list of object types available through the datastore.
Name	Specifies that the software search all object names for the search term. Specify whether the software finds an object with a name that Contains the string or Equals the string. Enter a string of characters to search for in the text box. If you choose to search in a repository, the casing is not applicable. If you choose to search in a datastore, and the name is case sensitive in that datastore: - Enter the search string as it appears in the database or application - Use double quotation marks around the string to preserve the case (" ").
Description	Specifies that the software search all object descriptions for the search term. Specify whether the software finds an object with a description that Contains the string or Equals the string. Enter a string of characters to search for in the text box. Any objects that are imported into the repository have a description from their source. Any object that you create has a description only if you added one when you created or edited the object.
Search all	Specifies that the software search every part of the object for the search string. You can only select Contains for this option. Enter a string of characters to search for in the text box. For jobs, the software searches in the job itself and every job element.
Advanced	Specifies to search for objects based on their attribute values. Applicable only when you select to look in the Repository.

3. Click [Search](#).

The software lists the objects that match your entries in the [Search](#) dialog box. A status line at the bottom of the [Search](#) dialog box shows where the software conducted the search (Local or Central), the total number of items found, and the amount of time it took to complete the search.

4. When you find a match in the results, drag the object name from the search results to the desired location, or right-click the object name and select one of the following options from the dropdown menu:
- [Open](#): Opens the object in the workspace.
 - [Import](#): Imports external tables from the specified datastore and stores the table metadata in the repository. Applicable for datastore type objects where you choose [External Data](#).
 - [Save as](#): Exports the search results to a CSV file. The software opens a [Save As](#) dialog box to select file name and location.
 - [View Where Used](#): Displays a list of parent objects in the [Output](#) Window.

- *Locate in library*: Displays the object in object library or the central library as applicable.
 - *Properties*: Opens the *Properties* dialog box for the object.
5. Click *Close* when you are done searching.

Task overview: [Objects \[page 52\]](#)

Related Information

[Reusable objects \[page 53\]](#)
[Single use objects \[page 54\]](#)
[Save reusable objects \[page 55\]](#)
[Object metadata \[page 57\]](#)
[Object descriptions \[page 58\]](#)
[Object hierarchy \[page 62\]](#)
[Object naming conventions \[page 65\]](#)
[Object editors \[page 66\]](#)
[Creating a reusable object in the object library \[page 67\]](#)
[Creating a reusable object using the tool palette \[page 68\]](#)
[Adding an existing object \[page 69\]](#)
[Changing object names \[page 70\]](#)
[Adding, changing, and viewing object properties \[page 71\]](#)
[Annotations \[page 74\]](#)
[Object descriptions \[page 75\]](#)
[Cutting or copying objects \[page 78\]](#)
[Replicating objects \[page 79\]](#)
[Save and delete objects \[page 80\]](#)
[Using View Where Used \[page 815\]](#)

8 Projects and Jobs

Project and job objects represent the top two levels of organization for the application flows you create using the Designer.

A project contains jobs that you create and execute.

A job is the only object that you execute. A job contains all other subordinate objects in hierarchical order.

[Projects \[page 86\]](#)

A project is a reusable object in which you group jobs.

[Jobs \[page 88\]](#)

A job is the only object in a project that you can execute.

8.1 Projects

A project is a reusable object in which you group jobs.

A project is the highest level of object organization in the software. Opening a project makes one group of objects easy to view and work with in Designer.

Use a project to group jobs that have interdependent schedules or jobs that you want to monitor together.

Projects have the following common characteristics:

- Listed in the [Projects](#) tab of the object library.
- Can only be open one at a time.
- Cannot be shared with multiple users.

[Objects that make up a project \[page 86\]](#)

The objects in a project appear hierarchically in the project area.

[Creating a new project \[page 87\]](#)

Create a project to group related jobs.

[Opening an existing project \[page 88\]](#)

An opened project appears in the project area of SAP Data Services Designer.

[Saving all changes to a project \[page 88\]](#)

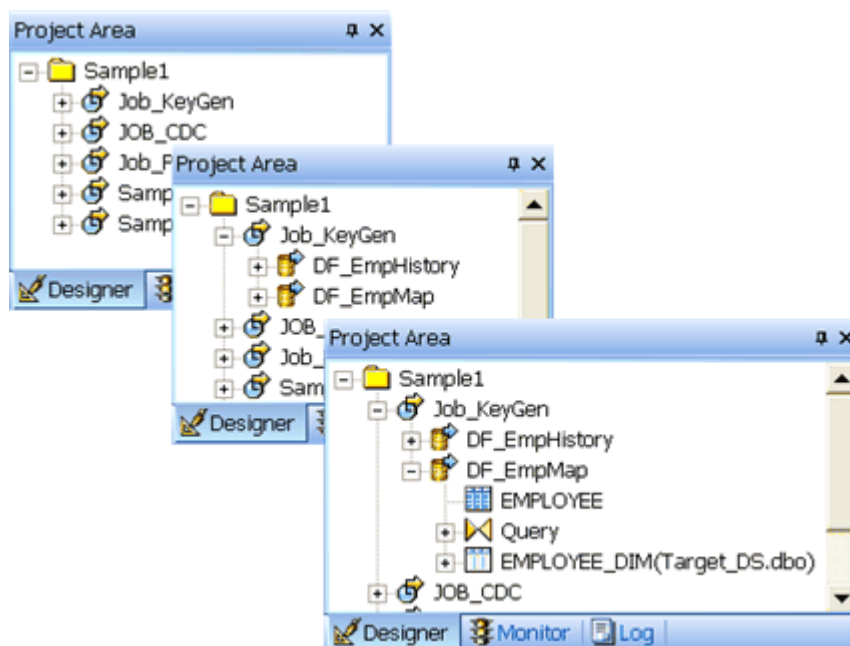
Save changes to objects as you create and edit objects, or save all changes when you exit the software.

8.1.1 Objects that make up a project

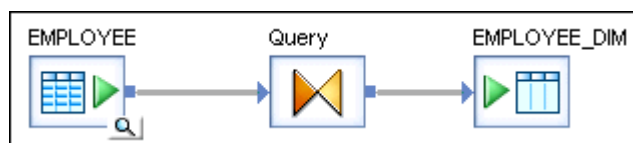
The objects in a project appear hierarchically in the project area.

The software lists a project in the project area in a file tree. Expand the project node to view the lower-level objects contained in the project. The lower-level objects may also be expandable based on the object type.

In the following example, the Job_KeyGen job contains two data flows, and the DF_EmpMap data flow contains multiple objects.



When you select an object in the project area, the software opens it in the workspace. The workspace displays work flow and data flow objects with icons in a flow diagram.



Related Information

[Project area \[page 26\]](#)

[Workspace \[page 31\]](#)

8.1.2 Creating a new project

Create a project to group related jobs.

1. Right-click in a blank area of the Project tab in the object library and select [New](#).

The software displays the [Project-New](#) dialog box that lists the name of your existing projects.

2. Enter the name of your new project in [Project name](#).

Enter a name using alphanumeric characters and underscores. The name cannot contain blank spaces.

3. Click [Create](#).

The new project appears in the project area and in the [Projects](#) tab of the object library.

8.1.3 Opening an existing project

An opened project appears in the project area of SAP Data Services Designer.

1. Open the [Projects](#) tab in the object library
2. Double-click the project to open it.
3. The project appears in the project area.

i Note

The software allows you to open only one project at a time. If another project is already open when you open a project, the software closes that project and opens the new one.

8.1.4 Saving all changes to a project

Save changes to objects as you create and edit objects, or save all changes when you exit the software.

To save changes during your software session:

1. Choose  [Project](#) > [Save All](#) .

The software displays the [Save all changes](#) dialog box that lists the jobs, work flows, and data flows that you edited since the last save.

2. If applicable, deselect any listed object that you do not want to save.
3. Click [OK](#).

i Note

The software also prompts you to save all changed objects when you execute a job and when you exit the Designer. Saving a reusable object also saves any single-use object that you included in the reusable object.

8.2 Jobs

A job is the only object in a project that you can execute.

When you work in your development environment, you manually execute and test jobs. When you move your jobs to a production environment, you can schedule batch jobs and set up real-time jobs as services. The jobs execute when the software receives a message request.

A job is made up of steps to execute together. Each step is represented by an object icon that you place in the workspace. You connect the steps to create a job diagram. A job diagram contains two or more objects connected together. Include any of the following objects in a job:

- Work flows, which contain:
 - Scripts
 - Conditionals
 - While Loops
 - Try/catch blocks
 - Data Flows
- Data flows, which contain:
 - Sources
 - Targets
 - Transforms

If a job becomes complex, organize the content into individual work flows, then create a single job that calls those work flows.

Real-time jobs use the same objects as batch jobs. You can add work flows and data flows to both batch and real-time jobs. When you drag a work flow or data flow icon into a job in the workspace, you are telling the software to validate these objects according to the requirements of the job type, batch or real time.

When you create real time jobs, be aware of restrictions regarding the use of some software features. For more information about real time job, see [Real-time Jobs \[page 730\]](#).

i Note

If a job using bulk load functionality fails, Data Services saves data files containing customer data in the Bulkload directory. You can then review and analyze the data. The default bulk load location is %DS_COMMON_DIR%/log/BulkLoader. For data protection and privacy concerns, ensure that you remove the files after analyzing them.

[Creating a job in the project area \[page 89\]](#)

Create a job in the project area to relate the job to the opened project.

[Creating a job in the object library \[page 90\]](#)

Create a job in the object library independent of a project.

[Naming conventions for objects in jobs \[page 90\]](#)

Adopting a naming convention for objects in your projects makes object identification easier.

Related Information

[Work flows \[page 632\]](#)

[Data flows \[page 271\]](#)

8.2.1 Creating a job in the project area

Create a job in the project area to relate the job to the opened project.

1. With the applicable project open, right-click in the project area and select either [New Batch Job](#) or [New Real-Time Job](#).

The software adds a new job node under the project name in the project area. The software opens the new job in the workspace pane.

2. Replace the default name in the text box with a new name for the job.

Enter a name using alphanumeric characters and underscores. You cannot use blank spaces in a job name.

3. Optional. Open the job properties to add a description of the job.

Next, build the new job by adding a work flow (optional), data flow, and other objects as applicable.

8.2.2 Creating a job in the object library

Create a job in the object library independent of a project.

1. Right-click in the *Jobs* tab of the object library and select either *Batch Job* or *Real-time Job*.

The software lists the new job with a default name under Batch Jobs or Real-time Jobs as applicable. The new job appears in alphabetical order based on the default name

2. While the default name is highlighted, enter a new name for the job in the object library.

Enter a name using alphanumeric characters and underscores. You cannot use blank spaces in the job name.

The software relists the new job in alphabetical order in the object library using the new name.

3. Right-click the job name and select *Properties*.

The *Properties* dialog box opens.

4. Enter a description for the job in the *Description* text box.
5. Click *OK* to close the *Properties* dialog box.

Add the new job to an open project by dragging the new job icon from the object library to the project area.

8.2.3 Naming conventions for objects in jobs

Adopting a naming convention for objects in your projects makes object identification easier.

We recommend that you follow consistent naming conventions to facilitate object identification across all systems in your enterprise. Using a consistent naming convention allows you to more easily identify object types when you work with objects in other applications such as:

- Data-modeling applications
- ETL applications
- Reporting applications
- Adapter software development kits

In the SAP Data Services documentation, we use the naming conventions detailed in the following table.

Prefix	Suffix	Object	Example
DF_		Data flow	DF_Currency
EDF_	_Input	Embedded data flow	EDF_Example_Input
EDF_	_Output	Embedded data flow	EDF_Example_Output
RTJob_		Real-time job	RTJob_OrderStatus
WF_		Work flow	WF_SalesOrg
JOB_		Job	JOB_SalesOrg
	_DS	Datastore	ORA_DS
DC_		Datastore configuration	DC_DB2_production
SC_		System configuration	SC_ORA_test
	_Memory_DS	Memory datastore	Catalog_Memory_DS
PROC_		Stored procedure	PROC_SalesStatus

Designer is a graphical user interface with icons representing objects in its windows. However, other interfaces might require you to identify object types by the text alone. By using prefix or suffix, you can more easily identify the object type.

In addition to prefixes and suffixes, you might want to provide standardized names for objects that identify a specific action across all object types. For example: DF_**OrderStatus**, RTJob_**OrderStatus**. You can also include path name identifiers in your naming conventions. For example, the stored procedure naming convention can look like either of the following:

- `<PREFIX>_<datastore>.<owner>.<ProcName>_<SUFFIX>`
`<PREFIX>_<datastore>.<owner>.<package>.<ProcName>_<SUFFIX>`

9 Datastores

Datastores contain connection configurations to databases and applications that contain your data.

SAP Data Services uses the connection information in the datastores to access metadata from a database or application and read from or write to that database or application while the software executes a job. Datastore configurations can be direct or through adapters.

After you create a datastore, import internal or external metadata to use for sources or targets in jobs.

Data Services datastores can connect to any of the following databases or applications:

- Databases and mainframe file systems.
- Applications that have prepackaged or user written adapters.
- J.D. Edwards One World and J.D. Edwards World, Oracle Applications, PeopleSoft, SAP Applications, SAP Data Quality Management, microservices for location data, Siebel Applications, and Google BigQuery. For more information, see the appropriate supplement guide.
- Remote servers using FTP, SFTP, and SCP.
- SAP systems: SAP Applications, BW Source, and BW Target.

i Note

The software reads and writes data stored in flat files through flat file formats instead of datastores. The software reads and writes data stored in XML documents through DTDs and XML Schemas instead of datastores.

The specific information that a datastore object can access depends on the connection configuration. When your database or application changes, ensure that you make corresponding changes in the datastore information in the software. The software does not automatically detect the new information.

i Note

Objects deleted from a datastore connection are identified in the project area and workspace by a red

"deleted" icon.  This visual flag allows you to find and update data flows affected by datastore changes.

One datastore can contain multiple configurations. Multiple configurations allow you to plan ahead for the different environments in which your datastore may be used. When you export the datastore from one environment to a different environment, the software includes all configurations. Therefore, multiple configurations ease the work involved when you migrate to new environments.

❖ Example

You have a datastore that you use in several jobs. The datastore has three configurations, DEV, TEST, and PROD. You use the datastore in several jobs in your DEV environment. However, you have just migrated all of those DEV jobs to your TEST environment. Instead of opening each applicable job to change the datastore configuration, you open the datastore and change the default configuration from DEV to TEST. All instances of the datastore now use the TEST configuration.

Group any set of datastore configurations into a system configuration. When running or scheduling a job, select a system configuration that contains the configurations for your current environment.

[Database datastores \[page 93\]](#)

Use database datastores to access supported database tables to use as sources and-or targets in data flows.

[Persistent cache datastores \[page 181\]](#)

A persistent cache datastore is a container for cache tables.

[Linked datastores \[page 186\]](#)

Linked datastores are datastores in a database link relationship.

[Adapter datastores \[page 190\]](#)

Adapter datastores provide access to application data through an adapter.

[Application datastores \[page 191\]](#)

Application datastores contain connection information to the specific application.

[Web service datastores \[page 194\]](#)

Web service datastores contain connection information to an external web service-based data source.

[Change a datastore definition \[page 195\]](#)

Change a datastore definition by editing the datastore options or properties.

[Create and manage multiple datastore configurations \[page 197\]](#)

Create multiple configurations in a datastore to group connections for similar sources or targets.

9.1 Database datastores

Use database datastores to access supported database tables to use as sources and-or targets in data flows.

Database datastores can represent single or multiple connections with the following databases:

- Legacy systems using Attunity Connect
- HP Vertica, IBM DB2, Informix, Microsoft SQL Server, MySQL, Netezza, Oracle, SAP ASE, SAP Data Federator, SAP SQL Anywhere, SAP HANA, SAP Vora, Sybase IQ, and Teradata databases using native connections, and more. See the current [Product Availability Matrix \(PAM\)](#) on the SAP Support Portal for an up-to-date list.
- Other databases through ODBC
- A repository using a memory datastore or persistent cache datastore

You can create a connection to most of the data sources using a server name connection (DSN-less) instead of the DSN (Data Source Name) or TNS (Transparent Network Substrate) name.

i Note

TNS is applicable for Oracle only. Therefore we use the term “DSN” unless we are specifically talking about Oracle.

DSN-less connections eliminate the need to configure the same DSN entries on every machine in a distributed environment, or to when you move the database to a different machine.

The following table tells you where you can find information about setting up and using a database datastore:

List of database datastores

	Database datastore
<i>Designer Guide</i>	Attunity
	Data Federator
	DB2
	Informix
	Memory
	Microsoft SQL
	MySQL
	Netezza
	ODBC
	Oracle
	Persistent cache
	SAP SQL Anywhere
	SAP ASE
	Sybase IQ
	Teradata
<i>Data Services Supplement for Big Data</i>	Amazon Redshift
	HP Vertica
	PostgreSQL
	SAP HANA
	Snowflake
	SAP Vora
	Hive
<i>Supplement for Hadoop</i>	Hive

[Defining a database datastore \[page 96\]](#)

Define at least one database datastore for each database or mainframe file system with which you exchange data.

[Datastore metadata \[page 102\]](#)

View datastore metadata in the datastore explorer or the *Datastore* tab of the object library.

[Imported metadata from database datastores \[page 107\]](#)

Import metadata for tables, functions, and stored procedures from database datastores.

[Attunity Connector \(mainframe datastore\) \[page 115\]](#)

Use the Attunity Connector datastore to access mainframe data sources through Attunity Connect.

[Data Federator \[page 121\]](#)

Use the Data Federator database datastore to access data for read only.

[IBM DB2 datastores \[page 124\]](#)

To use DB2 tables as sources or targets in SAP Data Services data flows, create an IBM DB2 database datastore and import tables.

[Informix \[page 127\]](#)

Create an Informix database datastore to use in Data Services.

[Microsoft SQL Server datastore \[page 129\]](#)

To access your Microsoft SQL Server data, create a Microsoft SQL Server database datastore and import data.

[Memory datastores \[page 136\]](#)

A memory datastore is a container for memory tables.

[MySQL datastore \[page 145\]](#)

To use MySQL data in your SAP Data Services processes, create a MySQL database datastore to access your MySQL data.

[ODBC \[page 161\]](#)

Create an ODBC database datastore to use in Data Services.

[Oracle datastores \[page 166\]](#)

To use Oracle data as sources and targets in data flows, create an Oracle database datastore.

[Netezza \[page 170\]](#)

Create a Netezza database datastore for Data Services.

[SAP SQL Anywhere \[page 172\]](#)

Create an SAP SQL Anywhere database datastore to use in Data Services.

[SAP ASE \[page 173\]](#)

To use SAP ASE metadata in SAP Data Services, create an SAP ASE database datastore.

[Sybase IQ \[page 174\]](#)

To view, browse, and import data from SAP IQ, create a database datastore.

[Teradata \[page 179\]](#)

Create a Teradata database datastore to use in Data Services.

Parent topic: [Datastores \[page 92\]](#)

Related Information

[Persistent cache datastores \[page 181\]](#)

[Linked datastores \[page 186\]](#)

[Adapter datastores \[page 190\]](#)

[Application datastores \[page 191\]](#)

[Web service datastores \[page 194\]](#)

[Change a datastore definition \[page 195\]](#)

[Create and manage multiple datastore configurations \[page 197\]](#)

[DSN-less and TNS-less connections](#)

9.1.1 Defining a database datastore

Define at least one database datastore for each database or mainframe file system with which you exchange data.

Before defining a database datastore, obtain the appropriate access privileges to the database or file system. Ensure that you have the correct ODBC or JDBC driver version, if applicable.

1. In Designer, right-click in an empty area of the [Datastores](#) tab in the object library and select [New](#) from the dropdown menu.
2. Enter the name of the new datastore in [Datastore name](#).

Use alphanumeric characters and underscores in the name. Do not use blank spaces in the name.

3. In the [Datastore type](#) list, select [Database](#).

After you select the datastore type, the software displays options relevant to that type. For more information about datastore options and for information about database types not discussed here, see the [Reference Guide](#).

The following table contains information about special situations related to database types:

Database type	Information
Amazon RDS for SQL Server 2017	To connect to Amazon RDS SQL Server 2017, set the Database Type to Microsoft and the Database Subtype to SQL Server on Premise .
Oracle	The Oracle database type supports TNS-less connections. To use a TNS-less connection, deselect Use TNS name and enter the host name, SID, and port.
Data Federator	Specify the catalog name and the schema name in the URL so that the imported data does not include all of the tables from each catalog. To specify the catalog name and the schema name in the URL, see Limiting imported Data Federator catalogs [page 124] .
SAP HANA	For SAP HANA versions earlier than 2.0 SPS 01 MDC, database datastores access a single database. For SAP HANA versions 2.0 SPS 01 MDC and later, database datastores access a defined tenant database.

4. Optional. When applicable, deselect [Use data source name](#) to enable DSN-less connections.

For Informix DSN-less connections, when Data Services is on a different computer than the Informix server, identify the Informix host following the steps in [Configuring DSN-less connection for Informix that uses a different server \[page 128\]](#).

5. To continue with DSN-less connection, enter information in the following options as applicable to your database: [Database Server Name](#), [Database Name](#), [Port](#), [User Name](#), and [Password](#).
6. Select [Enable automatic data transfer](#) when applicable to enable the Data_Transfer transform to use transfer tables in this datastore. The Data_Transfer transform pushes down subsequent database operations.

7. Click [Advanced](#).

i Note

The options in the [Advanced](#) section vary based on the type of database you select.

8. To convert a data type in your source that Data Services does not support, select [Import unsupported data types as VARCHAR of size](#) and enter the number of characters to allow.

For more information about data types, see the *Reference Guide*.

9. Click [OK](#) to save the database datastore.

[Basic and advanced configuration options \[page 98\]](#)

When you create a datastore, you start with basic settings and continue configuring the datastore using the advanced settings.

[Common datastore options \[page 99\]](#)

Most Advanced datastore options appear for specific datastore types, but many are common to more than one datastore type.

[Create multiple configurations per datastore \[page 101\]](#)

Additional datastore configurations enable you to configure more than one database type for the datastore connection.

[Change data capture \(CDC\) \[page 101\]](#)

CDC updates database files with data that changed since the last refresh.

[Enable Automatic Data Transfer \[page 102\]](#)

For automatic data transfer, the Data_Transfer transform uses transfer tables in the datastore to push down subsequent database operations for more efficient processing.

Task overview: [Database datastores \[page 93\]](#)

Related Information

[Datastore metadata \[page 102\]](#)

[Imported metadata from database datastores \[page 107\]](#)

[Attunity Connector \(mainframe datastore\) \[page 115\]](#)

[Data Federator \[page 121\]](#)

[IBM DB2 datastores \[page 124\]](#)

[Informix \[page 127\]](#)

[Microsoft SQL Server datastore \[page 129\]](#)

[Memory datastores \[page 136\]](#)

[MySQL datastore \[page 145\]](#)

[ODBC \[page 161\]](#)

[Oracle datastores \[page 166\]](#)

[Netezza \[page 170\]](#)

[SAP SQL Anywhere \[page 172\]](#)

[SAP ASE \[page 173\]](#)

[Sybase IQ \[page 174\]](#)

[Teradata \[page 179\]](#)

[Create and manage multiple datastore configurations \[page 197\]](#)

9.1.1.1 Basic and advanced configuration options

When you create a datastore, you start with basic settings and continue configuring the datastore using the advanced settings.

Settings for basic and advanced

Group of options	Descriptions
Basic datastore options	Basic options are in the initial dialog box of the datastore editor. The initial dialog box begins with three options: <i>Datastore Name</i>, <i>Datastore Type</i>, and <i>Database Type</i>. Additional options appear as you select the datastore type and the database type. Additional options can include security information, CDS (change data capture) options, DSN (data source name) options, user name, and password.
Advanced options	The advanced options are based on the datastore type and the database type. Advanced options can include FTP details, connection information, general loading information, and security information. Access advanced options by clicking <i>Advanced</i> in the configuration editor.

Parent topic: [Defining a database datastore \[page 96\]](#)

Related Information

[Common datastore options \[page 99\]](#)

[Create multiple configurations per datastore \[page 101\]](#)

[Change data capture \(CDC\) \[page 101\]](#)

[Enable Automatic Data Transfer \[page 102\]](#)

9.1.1.2 Common datastore options

Most Advanced datastore options appear for specific datastore types, but many are common to more than one datastore type.

The following table describes the datastore options that are common to multiple datastore types. For descriptions of options specific to the datastore type, see that datastore type topic.

Common option descriptions

Option	Description
Datastore Name	Enter a name for this datastore.
Datastore Type	Select the type of datastore to create.
Database Type	Select the database type that this datastore accesses.
Database version	Select the version of the database that this datastore accesses. Displayed options in the rest of the datastore editor vary depending on the version selected.
Use data source name (DSN)	Select to use DSN to connect to the database. When you select to use DSN, complete the Data source name option. By default, this option is not selected and the software uses a server name (also known as DSN-less) connection. For a DSN-less connection, complete the following options: • Database server name • Database name • Port
Data Source Name	Select or type the DSN defined in the ODBC Administrator for connecting to your database. This option is required when you select Use data source name (DSN).
ODBC Admin button	Opens the ODBC Administrator to create a DSN.
Database Server Name	Enter the name of the applicable database server. This option is required when you select Use data source name (DSN).
Hostname	Enter the name of the database defined in SAP IQ. This option is required if you did not select Use data source name (DSN).
Port	Enter the number of the database port. This option is required if you did not select Use data source name (DSN).
User name	Enter the user name of the account through which the software accesses the database.

Option	Description
Password	Enter the password associated with User name .
Additional connection parameters	Enter information for any additional parameters that the data source ODBC driver and database supports. Use the following format: <parameter1=value1; parameter2=value2>
Rows per commit	Enter the maximum number of rows loaded to a target table before saving the data. The default is 1000. This value is the default commit size for target tables in this datastore. You can overwrite this value for individual target tables.
Bulk loader directory	Enter or browse to the location where the software writes data files for bulk loading. You can use a variable for this option. i Note For Solaris systems, the path name must be less than 80 characters. The default location for the bulk loader directory is <DS_COMMON_DIR>/log/bulkload.
Overflow file directory	Enter or browse to the location where the software writes overflow files written by target tables in this datastore. You can use a variable for this option.
Language	For complete information, see the "Locales and Multi-byte Functionality" topic.
Code page	For complete information, see the "Locales and Multi-byte Functionality" topic.
Server code page	For complete information, see the "Locales and Multi-byte Functionality" topic.
Additional session parameters	Additional session parameters specified as a valid SQL statement. Delimit multiple SQL statements with semicolons.
Aliases	Click to open the Create New Alias popup dialog. Enter an alias name, click <i>OK</i> , and enter the owner name in the Alias text row.

Parent topic: [Defining a database datastore \[page 96\]](#)

Related Information

[Basic and advanced configuration options \[page 98\]](#)
[Create multiple configurations per datastore \[page 101\]](#)
[Change data capture \(CDC\) \[page 101\]](#)
[Enable Automatic Data Transfer \[page 102\]](#)
[Locales and Multi-byte Functionality](#)

9.1.1.3 Create multiple configurations per datastore

Additional datastore configurations enable you to configure more than one database type for the datastore connection.

At the bottom of the [Advanced](#) option section of the datastore editor, the software indicates the number of configurations that exist for the datastore. If you've just completed configuring a new datastore, the configuration number indicates 1 for the current configuration. To create more configurations, click [Edit](#) next to the configuration notation. The [Configurations for Datastore](#) dialog box opens.

The configuration editor contains a subset of the options in the datastore editor window. The new datastore is configuration 1. It appears in a column labeled Configuration1. Use the configuration editor to add, edit, and remove datastore configurations. Select one configuration as the default configuration.

For more information about creating multiple configurations, see the *Designer Guide*.

Parent topic: [Defining a database datastore \[page 96\]](#)

Related Information

[Basic and advanced configuration options \[page 98\]](#)
[Common datastore options \[page 99\]](#)
[Change data capture \(CDC\) \[page 101\]](#)
[Enable Automatic Data Transfer \[page 102\]](#)
[Create and manage multiple datastore configurations \[page 197\]](#)

9.1.1.4 Change data capture (CDC)

CDC updates database files with data that changed since the last refresh.

The software locates and extracts only data that changed since the last refresh, which can improve performance. Additionally, using CDC preserves history so you can track data changes over time.

The CDC option is active for specific data types and only for new database datastore creation.

Restrictions when you choose the CDC option:

- Any additional configurations for the datastore must be for the same database type.
- The datastore advanced option groups for *General*, *Bulk Loader*, and *FTP* are not available. That is because a CDC datastore is read-only so you can only use the datastore as a source in a data flow.

Parent topic: [Defining a database datastore \[page 96\]](#)

Related Information

[Basic and advanced configuration options \[page 98\]](#)

[Common datastore options \[page 99\]](#)

[Create multiple configurations per datastore \[page 101\]](#)

[Enable Automatic Data Transfer \[page 102\]](#)

9.1.1.5 Enable Automatic Data Transfer

For automatic data transfer, the Data_Transfer transform uses transfer tables in the datastore to push down subsequent database operations for more efficient processing.

The Enable automatic data transfer check box is selected by default when you create a new datastore and you choose *Database* for *Datastore type*. The check box displays for all databases except Attunity Connector, Data Federator, Memory, and Persistent Cache.

Parent topic: [Defining a database datastore \[page 96\]](#)

Related Information

[Basic and advanced configuration options \[page 98\]](#)

[Common datastore options \[page 99\]](#)

[Create multiple configurations per datastore \[page 101\]](#)

[Change data capture \(CDC\) \[page 101\]](#)

9.1.2 Datastore metadata

View datastore metadata in the datastore explorer or the *Datastore* tab of the object library.

Open a datastore to open the datastore explorer in the workspace. View external, internal, and repository metadata contained in the datastore source. You can perform the following actions in the datastore explorer:

- View and browse external tables
- View and browse imported tables
- Search for objects by name
- View whether the schema changed
- View secondary index information for a table
- Import metadata

After you import objects from a datastore, view and access the objects in the object library.

[Viewing datastore metadata \[page 104\]](#)

Datastore metadata includes a list of the objects available in the datastore.

[Viewing imported objects \[page 105\]](#)

After you import objects from the datastore, view the imported objects in your object library.

[Determining if a schema has changed since it was imported \[page 106\]](#)

SAP Data Services can indicate if the imported object has changed in the source since you last imported it.

[Viewing secondary index information for tables \[page 107\]](#)

View the secondary index information for an imported table to better understand the table schema.

Parent topic: [Database datastores \[page 93\]](#)

Related Information

[Defining a database datastore \[page 96\]](#)

[Imported metadata from database datastores \[page 107\]](#)

[Attunity Connector \(mainframe datastore\) \[page 115\]](#)

[Data Federator \[page 121\]](#)

[IBM DB2 datastores \[page 124\]](#)

[Informix \[page 127\]](#)

[Microsoft SQL Server datastore \[page 129\]](#)

[Memory datastores \[page 136\]](#)

[MySQL datastore \[page 145\]](#)

[ODBC \[page 161\]](#)

[Oracle datastores \[page 166\]](#)

[Netezza \[page 170\]](#)

[SAP SQL Anywhere \[page 172\]](#)

[SAP ASE \[page 173\]](#)

[Sybase IQ \[page 174\]](#)

[Teradata \[page 179\]](#)

9.1.2.1 Viewing datastore metadata

Datastore metadata includes a list of the objects available in the datastore.

1. In the *Datastores* tab of the object library, right-click a datastore and select *Open* from the dropdown menu.

The software opens the datastore in the datastore explorer in the workspace. The software uses the datastore connection information to open the connection to the source and display the metadata available.

2. Select *External metadata* to view tables in the external database.
3. Use the table columns in the datastore explorer to sort the objects by: *Metadata*, *Type*, *Imported*, or *Changed*.
4. Right-click an object such as a table to select an action from the dropdown menu.

The following table describes the external metadata dropdown menu options.

Command	Description
<i>Open</i>	Opens the table editor for the selected object. Lists table schema and contains information when applicable in the following tabs: ◦ <i>General</i> ◦ <i>Attributes</i> ◦ <i>Class Attributes</i> ◦ <i>Indexes</i> ◦ <i>Partitions</i> Applicable for tables only.
<i>Import</i>	Imports or reimports metadata from the selected object into your repository.
<i>Reconcile</i>	Compares the metadata in the selected object with the metadata of the object in your repository. Displays results in the <i>Changed</i> column: ◦ <i>Yes</i>: The software found a change between the current and imported object metadata. ◦ <i>No</i>: The software did not find a change between the current and imported object metadata. Available for imported objects only.

5. Select *Repository metadata* at the top of the explorer to view a list of imported tables.
6. Right-click an imported table and select an option from the dropdown menu.

The dropdown menu includes *Open*, *Import*, and *Reconcile* as in the external metadata menu, plus the additional options described in the following table.

Command	Description
Re-import	Reimports metadata from the database into the repository.  Caution Overwrites the repository version of the object with the database version of the object. All changes are overwritten.
Delete	Deletes the open table or tables from the repository.
Properties	Opens the Properties dialog box for the selected table.
View Data	Opens the View Data dialog box for the selected table, which contains a view of the data currently in the table. Also view the following applicable information: ◦ Profile tab ◦ Column Profile tab ◦ Related menu options for each tab

7. Close the datastore explorer by clicking the *X* in the upper right corner of the workspace.

Task overview: [Datastore metadata \[page 102\]](#)

Related Information

[Viewing imported objects \[page 105\]](#)

[Determining if a schema has changed since it was imported \[page 106\]](#)

[Viewing secondary index information for tables \[page 107\]](#)

9.1.2.2 Viewing imported objects

After you import objects from the datastore, view the imported objects in your object library.

1. Open the [Datastores](#) tab in the object library.
2. Expand the applicable datastore node.

The software lists all object types under the datastore node. The object type varies based on the type of datastore. For example, you may see tables, functions, and hierarchies.

3. Expand the node next to any of the objects listed.

The object node expands to show a list of imported objects for the object type. Some nodes may not expand if there are no imported objects for that type of object. For example, if you imported tables, the

Tables node expands to show a list of tables. However, if there are no imported functions, the Functions node does not expand.

4. You can use any datastore object in a data flow based on the type of object.

Task overview: [Datastore metadata \[page 102\]](#)

Related Information

[Viewing datastore metadata \[page 104\]](#)

[Determining if a schema has changed since it was imported \[page 106\]](#)

[Viewing secondary index information for tables \[page 107\]](#)

9.1.2.3 Determining if a schema has changed since it was imported

SAP Data Services can indicate if the imported object has changed in the source since you last imported it.

1. In the datastore explorer, select *Repository Metadata*.
2. Right-click the name of the applicable object and select *Reconcile* from the dropdown menu.

The software adds a *Yes* value in the *Changed* column when there has been a change. The software adds a *No* value in the *Changed* column when there has not been a change.

3. Optional. If the changed column indicates *Yes*, reimport the object to obtain the most up-to-date data.

The software overwrites the local version of the object with the source version of the object.

Caution

Keep in mind that you lose any modifications and changes that you have made to the object.

Task overview: [Datastore metadata \[page 102\]](#)

Related Information

[Viewing datastore metadata \[page 104\]](#)

[Viewing imported objects \[page 105\]](#)

[Viewing secondary index information for tables \[page 107\]](#)

9.1.2.4 Viewing secondary index information for tables

View the secondary index information for an imported table to better understand the table schema.

1. Open the [Datastores](#) tab in the object library.
2. Expand the Table node under the applicable datastore, right-click the applicable table name, and select [Properties](#) from the dropdown menu.

The [Properties](#) dialog box opens.

3. Open the [Indexes](#) tab.

The [Indexes](#) tab contains a table of Indexes on the left. The Indexes column contains the index name. The Unique column indicates a [Yes](#) or [No](#) based on whether the index is unique. The Clustered column indicates a [Yes](#) or [No](#) based on whether the index is clustered.

4. Click an index name to see a list of columns on the right.

Task overview: [Datastore metadata \[page 102\]](#)

Related Information

[Viewing datastore metadata \[page 104\]](#)

[Viewing imported objects \[page 105\]](#)

[Determining if a schema has changed since it was imported \[page 106\]](#)

9.1.3 Imported metadata from database datastores

Import metadata for tables, functions, and stored procedures from database datastores.

The software saves the table metadata that you import in the local repository. The software accesses the actual table data using the connection in the datastore. However, you can edit and configure the metadata when you use the tables and functions in your jobs.

[Imported table metadata \[page 108\]](#)

When you import tables through a database datastore, SAP Data Services imports a specific set of metadata information and stores the metadata in the repository.

[Imported stored function and procedure information \[page 110\]](#)

The software can import functions and stored procedures from a number of database management systems.

[Importing metadata by browsing \[page 111\]](#)

Import metadata from database datastores by browsing in the datastore explorer.

[Importing metadata by name \[page 112\]](#)

Import metadata from database datastores directly from object library.

[Importing metadata by searching \[page 113\]](#)

When you know that the name of an object to import contains a word or string, use search criteria to find the object for import.

[Reimport object metadata \[page 114\]](#)

Reimport objects when the source data has changed, or when you want to revert a changed object back to the original state.

Parent topic: [Database datastores \[page 93\]](#)

Related Information

[Defining a database datastore \[page 96\]](#)

[Datastore metadata \[page 102\]](#)

[Attunity Connector \(mainframe datastore\) \[page 115\]](#)

[Data Federator \[page 121\]](#)

[IBM DB2 datastores \[page 124\]](#)

[Informix \[page 127\]](#)

[Microsoft SQL Server datastore \[page 129\]](#)

[Memory datastores \[page 136\]](#)

[MySQL datastore \[page 145\]](#)

[ODBC \[page 161\]](#)

[Oracle datastores \[page 166\]](#)

[Netezza \[page 170\]](#)

[SAP SQL Anywhere \[page 172\]](#)

[SAP ASE \[page 173\]](#)

[Sybase IQ \[page 174\]](#)

[Teradata \[page 179\]](#)

9.1.3.1 Imported table metadata

When you import tables through a database datastore, SAP Data Services imports a specific set of metadata information and stores the metadata in the repository.

After you import metadata, you can edit metadata such as column names, descriptions, and data types. The edits that you make affect all objects that call the specific table.

Imported table metadata

Metadata	Description
Table name	Specifies the table name as it appears in the database. The maximum name length depends on the DBMS of your Data Services repository. Most database types have a maximum length for table names of 256 characters. Other database types are different. For example, MySQL is 64 characters and MS SQL Server is 128 characters.
Table description	The description of the table, if any.
Column name	The name of each column.
Column description	The description of each column, if any.
Column data type	The data type for each column. When the column contains a data type that Data Services does not support, Data Services converts the data type to a supported data type. The conversion happens only when you use the table in a work flow or data flow. In some cases, if the software cannot convert the data type, it ignores the column entirely.
Column content type	The content type for each column.
Primary key column	The column or columns that comprise the primary key for the table. The software shows primary key columns for tables used in a data flow with a key icon next to the column name.
Table attribute	Information such as the date the table was created and last modification date, when available.
Owner name	Name of the table owner. i Note The owner name for MySQL and Netezza data sources corresponds to the name of the database or schema where the table appears.

Parent topic: [Imported metadata from database datastores \[page 107\]](#)

Related Information

[Imported stored function and procedure information \[page 110\]](#)

[Importing metadata by browsing \[page 111\]](#)
[Importing metadata by name \[page 112\]](#)
[Importing metadata by searching \[page 113\]](#)
[Reimport object metadata \[page 114\]](#)
[Unsupported data types \[page 955\]](#)

9.1.3.2 Imported stored function and procedure information

The software can import functions and stored procedures from a number of database management systems.

The database types for which you can import stored procedures are: DB2, MS SQL Server, Oracle, SAP HANA, SQL Anywhere, SAP ASE, Sybase IQ, and Teradata databases. You can also import stored functions and packages from Oracle.

Use the imported functions and procedures and the extraction specifications you give Data Services in data flows.

Imported function metadata includes the following:

- Function parameters
- Return type
- Name, owner

Access the imported functions and procedures from the Function node listed under the datastore in the object library. When you use the function or stored procedure in a job, use the function wizard or smart editor as available. The software lists the function or stored procedure in the function wizard and smart editor under the category specified by the datastore.

Parent topic: [Imported metadata from database datastores \[page 107\]](#)

Related Information

[Imported table metadata \[page 108\]](#)
[Importing metadata by browsing \[page 111\]](#)
[Importing metadata by name \[page 112\]](#)
[Importing metadata by searching \[page 113\]](#)
[Reimport object metadata \[page 114\]](#)
[Reference Guide: About procedures \[page 787\]](#)

9.1.3.3 Importing metadata by browsing

Import metadata from database datastores by browsing in the datastore explorer.

i Note

You cannot import functions by browsing.

1. Open the [Datastores](#) tab in the object library.
2. Right-click the applicable datastore and choose [Open](#) from the dropdown menu.

The datastore explorer opens in the workspace.

3. Select [External Metadata](#) at the top.

The datastore explorer lists the objects available to import.

i Note

Objects listed when you select [Repository Metadata](#) have already been imported.

4. Expand any nodes if necessary to find the items to import.
5. Select the object or objects to import.
6. Right-click and select [Import](#) from the dropdown menu.
7. In the object library, go to the [Datastores](#) tab to display the list of imported objects.

The imported objects are available to select in the object library. To verify the import was successful:

1. Open the [Datastores](#) tab in the object library
2. Expand the applicable datastore node.
3. Expand the applicable object node.

The imported objects appear under the object.

Task overview: [Imported metadata from database datastores \[page 107\]](#)

Related Information

[Imported table metadata \[page 108\]](#)

[Imported stored function and procedure information \[page 110\]](#)

[Importing metadata by name \[page 112\]](#)

[Importing metadata by searching \[page 113\]](#)

[Reimport object metadata \[page 114\]](#)

9.1.3.4 Importing metadata by name

Import metadata from database datastores directly from object library.

1. Open the *Datastores* tab in the object library.
2. Right-click the applicable datastore name and choose *Import By Name* from the dropdown menu.

The *Import by Name* dialog box opens.

3. Choose the type of item you want to import from the *Type* dropdown list.

For database datastores, the options are *Table* and *Function*. To import a stored procedure, select *Function*.

4. To import tables, perform the following substeps:

- a. Enter the table name in the *Name* text box.

If the object name is case-sensitive in the database, and it is mixed case, enter the name as it appears in the database. Use double quotation marks (") around the name to preserve the case.

If the database type is Netezza 7.x, enter a schema name in the *Schema* text box to limit the specified tables to a particular schema. If you leave the schema name blank, the software limits the specified tables to the default schema.

- b. Select *All* to import all tables if applicable.

The *All* checkbox appears based on the database type.

- c. Enter an owner name in the *Owner* text box to limit the specified table or tables to a particular owner.

If you leave the *Owner* text box blank, the software finds all tables that match the other criteria regardless of owner.

5. To import functions or stored procedures, perform the following substeps:

- a. Enter the function or stored procedure name in the *Name* text box.

If the object name is case-sensitive in the database, and it is mixed case, enter the name as it appears in the database. Use double quotation marks (") around the name to preserve the case. If you do not use double quotation marks, the software converts names into all upper-case characters.

If the database type is Oracle, you can alternately enter the name of a package. The software allows you to import procedures or functions created within packages and use them as top-level procedures or functions. If you enter a package name, the software imports all stored procedures and stored functions defined within the Oracle package. You cannot import an individual function or procedure defined within a package.

- b. Enter an owner name in the *Owner* text box to limit the specified functions to a particular owner.

If you leave the owner name blank, the software finds function or stored procedure names regardless of owner.

If you are importing an Oracle function or stored procedure, clear the *Callable from SQL expression* checkbox under the following circumstance:

A stored procedure cannot be pushed down to a database inside another SQL statement when the stored procedure contains:

- A DDL statement
- Ends the current transaction with COMMIT or ROLLBACK
- Issues any ALTER SESSION or ALTER SYSTEM commands

6. Click *OK*.

Task overview: [Imported metadata from database datastores \[page 107\]](#)

Related Information

[Imported table metadata \[page 108\]](#)

[Imported stored function and procedure information \[page 110\]](#)

[Importing metadata by browsing \[page 111\]](#)

[Importing metadata by searching \[page 113\]](#)

[Reimport object metadata \[page 114\]](#)

9.1.3.5 Importing metadata by searching

When you know that the name of an object to import contains a word or string, use search criteria to find the object for import.

i Note

Functions cannot be imported by searching.

1. Open the [Datastores](#) tab in the object library.
2. Right-click the name of the applicable datastore and select [Search](#) from the dropdown menu.

The [Search](#) dialog box opens. Data Services automatically populates the [Look in](#) text box with the name of the datastore.
3. Select [External Data](#) from the dropdown list.
 - External indicates that the software searches for the item in the entire database defined by the datastore.
 - Internal indicates that the software searches only the items that have been imported.
4. Select [Local Repository](#) or [Central Repository](#) as applicable.

[Central Repository](#) is available when you have a central repository.
5. Select [Tables](#) from the [Object Type](#) dropdown list.

i Note

The software only imports tables when you use the search method.

6. Enter search criteria for [Name](#), [Description](#), and [Search all](#) options as applicable.
7. Select [Contains](#) or [Equals](#) from the dropdown list based on whether you provide a complete or partial search value.

For [Name](#), when you select the [Equals](#) criteria, enter the full search string. For example, enter `owner.table_name` rather than just `table_name`.
8. Complete criteria in the [Advanced](#) tab to search using the object attribute values.

The advanced options only apply to searches of items that you have already imported.

9. Click [Search](#).

The software lists the objects that match your search criteria.

10. To import an object from the returned list, right-click the object name and choose [Import](#) from the dropdown list.

Task overview: [Imported metadata from database datastores \[page 107\]](#)

Related Information

[Imported table metadata \[page 108\]](#)

[Imported stored function and procedure information \[page 110\]](#)

[Importing metadata by browsing \[page 111\]](#)

[Importing metadata by name \[page 112\]](#)

[Reimport object metadata \[page 114\]](#)

9.1.3.6 Reimport object metadata

Reimport objects when the source data has changed, or when you want to revert a changed object back to the original state.

Reimporting overwrites any changes you might have made to the object metadata in the software. You can reimport objects that you used in previous versions of the software. Use the object library at the following levels to reimport objects from previous versions of the software:

- Individual objects: Reimport the metadata for an individual object such as a table or function
- Category node level: Reimport the definitions of all objects of that type in that datastore. For example, right-click the Tables node and select [Import](#) to reimport all tables from the datastore.
- Datastore level: Reimport the entire datastore and all its dependent objects including tables, functions, IDocs, and hierarchies.

If using Netezza 6.x or 7.x, the software reimports tables that were upgraded from SP9 or below without a schema name qualifier. For Netezza 7.x, the software reimports tables that were upgraded and imported into SP10 with the schema name showing.

Parent topic: [Imported metadata from database datastores \[page 107\]](#)

Related Information

[Imported table metadata \[page 108\]](#)

[Imported stored function and procedure information \[page 110\]](#)

[Importing metadata by browsing \[page 111\]](#)

[Importing metadata by name \[page 112\]](#)

[Importing metadata by searching \[page 113\]](#)

9.1.3.6.1 Reimporting objects from the object library

Reimport datastore objects from the object library.

Use the [View Where Used](#) feature to display where an object is currently being used. This helps you decide whether to reimport metadata and overwrite your existing data.

1. Open the [Datastores](#) tab in the object library.
2. Right-click an individual object and click [Reimport](#) from the dropdown menu.
3. Alternately, right-click a category node or datastore name and click [Reimport All](#).

You can also select multiple individual objects using Ctrl-click or Shift-click.

4. Click [Yes](#) to the popup message to reimport the metadata.

If you selected multiple objects to reimport and used [Reimport All](#), the software requests confirmation for each object unless you check the box [Don't ask me again for the remaining objects](#).

9.1.4 Attunity Connector (mainframe datastore)

Use the Attunity Connector datastore to access mainframe data sources through Attunity Connect.

The Attunity datastore accesses the following data sources:

- Adabas
- DB2 UDB for OS/390 and DB2 UDB for OS/400
- IMS/DB
- VSAM
- Flat files on OS/390 and flat files on OS/400

For a complete list of sources, refer to your Attunity documentation.

[Prerequisites for an Attunity datastore \[page 116\]](#)

Before you create an Attunity datastore, install separate software that uses the ODBC interface to connect to Attunity Connector.

[Creating and configuring an Attunity datastore \[page 117\]](#)

Create an Attunity datastore in Designer using the Datastore editor.

[Attunity Connector datastore options \[page 118\]](#)

Create an Attunity Connector database datastore for use with main frame systems only.

[Specifying multiple data sources in one Attunity datastore \[page 119\]](#)

The Attunity Connector datastore allows access to enter multiple Attunity database types on the same Attunity daemon location.

[Data Services naming convention for Attunity tables \[page 120\]](#)

Because a single datastore can access multiple software systems that do not share the same namespace, specify the name of the Attunity data source when you refer to a table.

[Limitations for Attunity datastore \[page 121\]](#)

Use almost all Data Services features with an Attunity Connector datastore with a few exceptions.

Parent topic: [Database datastores \[page 93\]](#)

Related Information

[Defining a database datastore \[page 96\]](#)

[Datastore metadata \[page 102\]](#)

[Imported metadata from database datastores \[page 107\]](#)

[Data Federator \[page 121\]](#)

[IBM DB2 datastores \[page 124\]](#)

[Informix \[page 127\]](#)

[Microsoft SQL Server datastore \[page 129\]](#)

[Memory datastores \[page 136\]](#)

[MySQL datastore \[page 145\]](#)

[ODBC \[page 161\]](#)

[Oracle datastores \[page 166\]](#)

[Netezza \[page 170\]](#)

[SAP SQL Anywhere \[page 172\]](#)

[SAP ASE \[page 173\]](#)

[Sybase IQ \[page 174\]](#)

[Teradata \[page 179\]](#)

9.1.4.1 Prerequisites for an Attunity datastore

Before you create an Attunity datastore, install separate software that uses the ODBC interface to connect to Attunity Connector.

Attunity Connector accesses mainframe data using software that you manually install on the mainframe server and the local client (Job Server) computer. The software connects to Attunity Connector using its ODBC interface.

It is not necessary to purchase a separate ODBC driver manager for UNIX and Windows platforms.

Servers

Install and configure the Attunity Connect product on the server, such as a zSeries computer.

Clients

To access mainframe data using Attunity Connector, install the Attunity Connect product. The ODBC driver is required. Attunity also offers an optional tool called Attunity Studio for configuration and administration.

Configure ODBC data sources on the client, which is SAP Data Services Job Server.

When you install a Job Server on UNIX, the installer prompts you to provide an installation directory path for Attunity connector software. In addition, you do not install a driver manager, because the software loads ODBC drivers directly on UNIX platforms.

For more information about how to install and configure these products, refer to the product documentation for the specific product.

Parent topic: [Attunity Connector \(mainframe datastore\) \[page 115\]](#)

Related Information

[Creating and configuring an Attunity datastore \[page 117\]](#)

[Attunity Connector datastore options \[page 118\]](#)

[Specifying multiple data sources in one Attunity datastore \[page 119\]](#)

[Data Services naming convention for Attunity tables \[page 120\]](#)

[Limitations for Attunity datastore \[page 121\]](#)

9.1.4.2 Creating and configuring an Attunity datastore

Create an Attunity datastore in Designer using the Datastore editor.

To use the Attunity Connector datastore option, upgrade your repository to SAP Data Services version 6.5.1 or later.

1. In the [Datastores](#) tab of the object library, right-click and select [New](#).
2. Enter a name for the datastore.
3. Select [Database](#) from the [Datastore type](#) dropdown list.
4. Select [Attunity Connector](#) from the [Database type](#) dropdown list.
5. Enter the following information in the applicable options:
 - Attunity data source name
 - Location of the Attunity daemon in [Host location](#)
 - Attunity daemon port number
 - Unique Attunity server workspace name
6. To change any of the default options (such as [Rows per Commit](#) or [Language](#)), click the [Advanced](#) button.
7. Click [OK](#).

You can now use the new datastore connection to import metadata tables into the current repository.

Task overview: [Attunity Connector \(mainframe datastore\) \[page 115\]](#)

Related Information

[Prerequisites for an Attunity datastore \[page 116\]](#)

[Attunity Connector datastore options \[page 118\]](#)

[Specifying multiple data sources in one Attunity datastore \[page 119\]](#)

[Data Services naming convention for Attunity tables \[page 120\]](#)

[Limitations for Attunity datastore \[page 121\]](#)

[Attunity Connector datastore options \[page 118\]](#)

9.1.4.3 Attunity Connector datastore options

Create an Attunity Connector database datastore for use with main frame systems only.

The following tables describe the options specific for Attunity Connector when you create or edit a datastore.

Main window

Attunity option	Description
<i>CDC options</i>	<i>No CDC</i> : Select to not use Change Data Capture. <i>Native CDC</i> : Select to use Attunity Change Data Capture.
<i>Data source name</i>	Enter the Attunity data source name or names as defined in Attunity Studio. Separate multiple data source names with semicolons.
<i>Host location</i>	Enter the computer name, fully qualified domain name, or IP address of the Attunity server computer (host).
<i>Attunity workspace</i>	Enter the workspace name under which the data sources are defined in Attunity Studio. For information, refer to your database requirements.

Parent topic: [Attunity Connector \(mainframe datastore\) \[page 115\]](#)

Related Information

[Prerequisites for an Attunity datastore \[page 116\]](#)

[Creating and configuring an Attunity datastore \[page 117\]](#)

[Specifying multiple data sources in one Attunity datastore \[page 119\]](#)

[Data Services naming convention for Attunity tables \[page 120\]](#)

[Limitations for Attunity datastore \[page 121\]](#)

[Change data capture \(CDC\) \[page 101\]](#)

9.1.4.4 Specifying multiple data sources in one Attunity datastore

The Attunity Connector datastore allows access to enter multiple Attunity database types on the same Attunity daemon location.

If you have several types of data on the same computer, for example a DB2 database and VSAM data sets, you might want to access data from both types using a single connection. For example, you can use a single connection to join tables and push the join operation down to a remote server, which reduces the amount of data transmitted through your network.

Before you list multiple data source names for one Attunity Connector datastore, ensure that you meet the following requirements:

- You access all Attunity data sources with the same user name and password.
 - You work with all Attunity data sources in the same workspace. When you set up access to the data sources in Attunity Studio, use the same workspace name for each data source.
1. Create a new or edit an existing Attunity Connector datastore in Designer.
 2. Complete configuration options as applicable.
 3. Enter multiple data sources in the [Data Source Name](#) text box and separate each name using semicolons.

❖ Example

Use the following format to enter multiple data sources:

```
AttunityDataSourceName;AttunityDataSourceName
```

❖ Example

For example, if you have a DB2 data source named DSN4 and a VSAM data source named Navdemo, enter the following values into the Data source box:

```
DSN4;Navdemo
```

4. Save the datastore.

Task overview: [Attunity Connector \(mainframe datastore\) \[page 115\]](#)

Related Information

[Prerequisites for an Attunity datastore \[page 116\]](#)

[Creating and configuring an Attunity datastore \[page 117\]](#)

[Attunity Connector datastore options \[page 118\]](#)

[Data Services naming convention for Attunity tables \[page 120\]](#)

[Limitations for Attunity datastore \[page 121\]](#)

[Attunity Connector datastore options \[page 118\]](#)

9.1.4.5 Data Services naming convention for Attunity tables

Because a single datastore can access multiple software systems that do not share the same namespace, specify the name of the Attunity data source when you refer to a table.

With an Attunity Connector, precede the table name with the data source and owner names separated by a colon. The format is as follows:

```
AttunityDataSource:OwnerName.TableName
```

When using the Designer to create your jobs with imported Attunity tables, Data Services automatically generates the correct SQL with the table name format. However, when you author SQL, be sure to use the table name format. You can author SQL in the following constructs:

- SQL function
- SQL transform
- Pushdown_sql function
- Preload commands in table loader
- Post-load commands in table loader

i Note

For tables in Data Services, the maximum length of the owner name for most repository types is 256. MySQL is 64 and MS SQL server is 128. For Attunity tables, the maximum length of the Attunity data source name and actual owner name is 63. The colon accounts for 1 character.

Parent topic: [Attunity Connector \(mainframe datastore\) \[page 115\]](#)

Related Information

[Prerequisites for an Attunity datastore \[page 116\]](#)

[Creating and configuring an Attunity datastore \[page 117\]](#)

[Attunity Connector datastore options \[page 118\]](#)

[Specifying multiple data sources in one Attunity datastore \[page 119\]](#)

[Limitations for Attunity datastore \[page 121\]](#)

9.1.4.6 Limitations for Attunity datastore

Use almost all Data Services features with an Attunity Connector datastore with a few exceptions.

When you use the Attunity Connector datastore, there are some Data Services features that are not available for processing:

- Bulk loading
- Imported functions for metadata tables
- Template tables used for creating tables
- The datetime data type supports up to 2 subseconds only
- Loading timestamp data into a timestamp column in a table

i Note

Attunity truncates varchar data to 8 characters, which is not enough to correctly represent a timestamp value.

Additionally, when you run a job that uses the Attunity Connector data on UNIX, the job could fail with the following error: [D000] Cannot open file /usr1/attun/navroot/def/sys System error 13: The file access permissions do not allow the specified action.; (OPEN)

This error occurs because of insufficient file permissions to some of the files in the Attunity installation directory. To avoid this error, change the file permissions for all files in the Attunity directory to 777 by executing the following command from the Attunity installation directory: `$ chmod -R 777 *`

Parent topic: [Attunity Connector \(mainframe datastore\) \[page 115\]](#)

Related Information

[Prerequisites for an Attunity datastore \[page 116\]](#)

[Creating and configuring an Attunity datastore \[page 117\]](#)

[Attunity Connector datastore options \[page 118\]](#)

[Specifying multiple data sources in one Attunity datastore \[page 119\]](#)

[Data Services naming convention for Attunity tables \[page 120\]](#)

9.1.5 Data Federator

Use the Data Federator database datastore to access data for read only.

The following tables contain descriptions for datastore options specific for Data Federator.

Advanced option descriptions

Option	Description
<i>General</i>	
<i>Database server working directory</i>	Enter the working directory for the load utility on the computer that runs the Data Federator server. Complete this field whenever the Data Federator server and the Job Server run on separate machines.
<i>FTP</i>	
<i>FTP hostname</i>	Enter the computer name, fully qualified domain name, or IP address. If this field is left blank or contains the name of the computer (host) where the Job Server resides, the software assumes that Data Federator and the software share the same computer and that FTP is unnecessary. When FTP is unnecessary, all other FTP-related fields can remain blank.
<i>FTP login user name</i>	Enter the FTP user name. Required to use FTP.
<i>FTP login password</i>	Enter the password related to the FTP logon user name. Required to use FTP.
<i>FTP host working directory</i>	Enter the absolute file path to the location on the database server where Data Services transfers the data file between the Job Server and the Data Federator server. For Windows servers, enter the virtual file path
<i>ODBC Miscellaneous</i>	
<i>Date format</i>	Enter a date format that Data Federator and the ODBC driver supports. The default date format in Data Services is <code>yyyy-mm-dd</code>.
<i>Time format</i>	Enter a time format that Data Federator and the ODBC driver supports. The default time format in Data Services is <code>hh:mi:ss</code>.
<i>Date-time format</i>	Enter a date/time format that Data Federator and the ODBC driver supports. The default date/time format in Data Services is <code>yyyy-mm-dd hh:mi:ss</code>.
<i>Decimal separator</i>	Enter the character that Data Federator uses to separate the decimal portion of a number. The default decimal separator for Data Services is a period.

Option	Description
<i>Data type conversion support</i>	Select an explicit convert function call that the software generates when there is a data type mismatch in an expression. The default is Automatic. Values include: • Automatic • ODBC syntax • SQL-92 syntax • No: Disables this option.
<i>NVL support</i>	Select the value to use when the input value is NULL. The default is Automatic. Values include: • Automatic • ODBC syntax • No: Disables this option.
<i>Ifthenelse support</i>	Yes: Allows conditional logic in mapping and selection operations. No: Does not allow conditional logic in mapping and selection operations.

[Limiting imported Data Federator catalogs \[page 124\]](#)

Limit Data Federator tables so that you do not import tables from all catalogs.

Parent topic: [Database datastores \[page 93\]](#)

Related Information

[Defining a database datastore \[page 96\]](#)

[Datastore metadata \[page 102\]](#)

[Imported metadata from database datastores \[page 107\]](#)

[Attunity Connector \(mainframe datastore\) \[page 115\]](#)

[IBM DB2 datastores \[page 124\]](#)

[Informix \[page 127\]](#)

[Microsoft SQL Server datastore \[page 129\]](#)

[Memory datastores \[page 136\]](#)

[MySQL datastore \[page 145\]](#)

[ODBC \[page 161\]](#)

[Oracle datastores \[page 166\]](#)

[Netezza \[page 170\]](#)

[SAP SQL Anywhere \[page 172\]](#)

[SAP ASE \[page 173\]](#)

[Sybase IQ \[page 174\]](#)

[Teradata \[page 179\]](#)

[Common datastore options \[page 99\]](#)

[ODBC datastore option value descriptions \[page 164\]](#)

9.1.5.1 Limiting imported Data Federator catalogs

Limit Data Federator tables so that you do not import tables from all catalogs.

To limit the tables to tables in a specific catalog, specify the catalog name and the schema name in the URL. In the datastore editor for Data Federator:

1. Select *ODBC Admin* and then the *System DSN* tab.
2. Highlight *Data Federator* and then click *Configure*.
3. In the URL option, enter the catalog name and the schema name. For example: `jdbc:select://localhost/<catalogname>;schema=<schemaname>`

Task overview: [Data Federator \[page 121\]](#)

9.1.6 IBM DB2 datastores

To use DB2 tables as sources or targets in SAP Data Services data flows, create an IBM DB2 database datastore and import tables.

Before you create a DB2 database datastore, ensure that you perform the following tasks:

- Obtain a supported ODBC driver, and configure it using either the ODBC Data Source Administrator (Windows) or the SAP Data Services Connection Manager (UNIX and Linux).
- If you use SSL/TLS protocol for encrypting SSL and your server authentication, or the newer Transport Layer Security (TLS), obtain the SSL certificate and key files as directed by your DB2 documentation.
- If you use SSL, ensure that you configure your environment as detailed in your IBM DB2 documentation for server side and client side SSL support.

Determine whether to configure a datastore using a data source name (DSN) or server-based (DSN-less) connection. Be aware of the following restrictions:

- SSL/TLS applies to DSN-less connections only.
- When you design data flows with bulk loading for a DB2 DSN-less connection, Data Services requires that you select the CLI Load mode. The Import mode is not applicable for DB2 DSN-less connections.

The following table describes datastore configuration options specific to DB2.

Main window

Option	Description
No CDC	Specifies whether to use change data capture (CDC). Replication Server CDC: Uses the Replication Server for change data capture. i Note Replication Server CDC is available for DB2 UDB 9.x and 10.x. It is not available for DB2 UDB 11.x. No CDC: Disables CDC processing. No CDC is the default setting, and it is the only value available for DB2 UDB 11.x.
Use data source name (DSN)	Specifies whether to use a DSN connection. If you select this option, the ODBC Admin... button appears. To create a DSN, click ODBC Admin... to open the ODBC Data Source Administrator. Do not select to use a DSN connection under the following circumstances: - You plan to use a server-named (DSN-less) connection. - You plan to use SSL/TLS protocol for secure network communication.
The following options appear when you do not select Use data source name (DSN) :	
Database server name Database name Port User Name Password	Specifies the server on which the database resides, the database to connect to, the port number, and the user name and password to access the database.
The following options appear when you select Use data source name (DSN) :	
Data source name	Select the DSN you created for this database from the drop-down list.
User name Password	Specifies the user name and password to access the DSN.

The following table contains options that apply to DB2 when you click [Advanced](#).

Option	Description
Use SSL encryption	Specifies to use SSL/TLS encryption for this datastore. Applicable only for DSN-less connections.

Option	Description
Encryption parameters	Opens the Encryption Parameters dialog box. To open the Encryption Parameters dialog box, click in the option cell and click the "..." that appears at the end, or double-click in the cell. Enabled only when you create a DSN-less connection and you select Yes for Use SSL encryption.
The following options appear in the Encryption Parameters dialog box:	
Key Store Database	Specifies the location and file name for your key store database, which contains server certificates. The key store database has the file extension <code>KDB</code> . For example, <code>mydbclient.kdb</code> .
Key Stash	Specifies the location and file name for the stash file, which the GSKit uses to obtain the password to the key store database. The stash file has the extension <code>STH</code> . For example, <code>mydbclient.sth</code> .
Complete the following options to configure bulk loading to the target table. For complete information about bulk loading, see the Performance Optimization Guide.	
Bulk loader user name	Specifies the name of the user authorized for bulk loading. For example, the user should have import and load permissions. Note When you create a DSN-less connection for the DB2 database datastore, the only bulk load mode available to you is the CLI Load mode. You select the mode in the target editor in your data flow.
Bulk loader password	Specifies the password for the user that you specified in Bulk loader user name .
DB2 server working directory	Specifies the working directory for the load utility on the computer that runs the DB2 server. Complete this option whenever the DB2 server and the Job Server run on separate machines.
The following options are for FTP.	
FTP host name	Specifies the FTP host name. Enter the computer name, fully qualified domain name, or IP address of the FTP host. If you leave this field blank or it contains the name of the computer (host) where the Job Server resides, the software assumes that DB2 and the software share the same computer and that FTP is unnecessary. When FTP is unnecessary, leave all other FTP-related fields blank.
FTP login user name	Specifies the name of the user defined to use FTP.
FTP login password	Specifies the password of the user defined to use FTP.

Parent topic: [Database datastores \[page 93\]](#)

Related Information

[Defining a database datastore \[page 96\]](#)
[Datastore metadata \[page 102\]](#)
[Imported metadata from database datastores \[page 107\]](#)
[Attunity Connector \(mainframe datastore\) \[page 115\]](#)
[Data Federator \[page 121\]](#)
[Informix \[page 127\]](#)
[Microsoft SQL Server datastore \[page 129\]](#)
[Memory datastores \[page 136\]](#)
[MySQL datastore \[page 145\]](#)
[ODBC \[page 161\]](#)
[Oracle datastores \[page 166\]](#)
[Netezza \[page 170\]](#)
[SAP SQL Anywhere \[page 172\]](#)
[SAP ASE \[page 173\]](#)
[Sybase IQ \[page 174\]](#)
[Teradata \[page 179\]](#)
[Common datastore options \[page 99\]](#)
[Change data capture \(CDC\) \[page 101\]](#)
[Bulk Loading in IBM DB2 Universal Database
Using the Connection Manager
Using the ODBC Drivers Selector for Windows](#)

9.1.7 Informix

Create an Informix database datastore to use in Data Services.

i Note

To use large object types with Informix, enable large object support when you create the DSN with the Informix driver.

- For Windows, open the ODBC Administrator, select the Informix driver, and click [Configure](#). The *Informix ODBC Driver Setup* dialog box opens. Select *Report Standard ODBC Types only* to enable large object support.
- For Unix, edit the `odbc.ini` file to set the option `NEEDODBCTYPESONLY=1`.

For more information about Informix ODBC driver configuration, see your Informix ODBC driver configuration documentation.

[Configuring DSN-less connection for Informix that uses a different server \[page 128\]](#)

When your Data Services installation is on a different computer than your Informix server, follow a different procedure to configure a DSN-less connection.

Parent topic: [Database datastores \[page 93\]](#)

Related Information

[Defining a database datastore \[page 96\]](#)
[Datastore metadata \[page 102\]](#)
[Imported metadata from database datastores \[page 107\]](#)
[Attunity Connector \(mainframe datastore\) \[page 115\]](#)
[Data Federator \[page 121\]](#)
[IBM DB2 datastores \[page 124\]](#)
[Microsoft SQL Server datastore \[page 129\]](#)
[Memory datastores \[page 136\]](#)
[MySQL datastore \[page 145\]](#)
[ODBC \[page 161\]](#)
[Oracle datastores \[page 166\]](#)
[Netezza \[page 170\]](#)
[SAP SQL Anywhere \[page 172\]](#)
[SAP ASE \[page 173\]](#)
[Sybase IQ \[page 174\]](#)
[Teradata \[page 179\]](#)
[Common datastore options \[page 99\]](#)

9.1.7.1 Configuring DSN-less connection for Informix that uses a different server

When your Data Services installation is on a different computer than your Informix server, follow a different procedure to configure a DSN-less connection.

Identify the Informix host as follows:

1. Go to your Informix client installation folder. For example: `C:\Program Files\IBM\Informix\Client-SDK\bin` Run `setnet32.exe`.
2. In the [Server Information](#) tab, enter the name of the IBM Informix server, the host name, and other required information.
3. Make the IBM Informix server the default server.

Designer obtains the Informix host name for the Informix server name you provided.

Task overview: [Informix \[page 127\]](#)

9.1.8 Microsoft SQL Server datastore

To access your Microsoft SQL Server data, create a Microsoft SQL Server database datastore and import data.

For Windows, use the bundled SQL Server Native Client ODBC driver, which is bundled with your SQL Server client installation. For Unix/Linux, use the DataDirect driver for SQL Server, which is bundled with your SAP Data Services installation.

Note

Data Services installation bundles certain DataDirect ODBC drivers. If there is a problem when processing Microsoft SQL Server data using a UNIX Job Server, the software issues an error message that might contain a DataDirect internal error message. DataDirect error-message syntax resembles the following structure: `<vendor-ODBCcomponent: message>`. See the documentation on the DataDirect Web site for more information about DataDirect message text or codes. For information about configuring DataDirect ODBC for Data Services, see the topic “Configuring DataDirect ODBC” under the Server Management section in the *Administrator Guide*.

As an option, include SSL/TLS protocol when you create the datastore. With SSL/TLS enabled, Data Services validates the SSL server certificate when you process jobs using data from the datastore.

Optionally enable change data capture (CDC) and bulk loading in the Microsoft SQL Server datastore configuration. Make sure that the Microsoft SQL Server version you select supports change data capture and bulk loading.

[Microsoft SQL Server database datastore options \[page 130\]](#)

When you create a Microsoft SQL Server database datastore, consider the purpose of the datastore and make sure that you complete all prerequisites for that particular purpose.

[Importing SSL certificate from Microsoft SQL Server \[page 136\]](#)

If you plan to connect to SQL Server using SSL encryption and validating the SSL certificate, you must first import the SSL certificate from Microsoft SQL Server.

Parent topic: [Database datastores \[page 93\]](#)

Related Information

[Defining a database datastore \[page 96\]](#)

[Datastore metadata \[page 102\]](#)

[Imported metadata from database datastores \[page 107\]](#)

[Attunity Connector \(mainframe datastore\) \[page 115\]](#)

[Data Federator \[page 121\]](#)

[IBM DB2 datastores \[page 124\]](#)

[Informix \[page 127\]](#)

[Memory datastores \[page 136\]](#)

[MySQL datastore \[page 145\]](#)

[ODBC \[page 161\]](#)

[Oracle datastores \[page 166\]](#)

[Netezza \[page 170\]](#)

[SAP SQL Anywhere \[page 172\]](#)

[SAP ASE \[page 173\]](#)

[Sybase IQ \[page 174\]](#)

[Teradata \[page 179\]](#)

9.1.8.1 Microsoft SQL Server database datastore options

When you create a Microsoft SQL Server database datastore, consider the purpose of the datastore and make sure that you complete all prerequisites for that particular purpose.

For example, you can create a datastore for capturing changed data, or you may want to enable bulk loading into the target table of a data flow. The following table includes options for the special purposes of a Microsoft SQL Server datastore.

Main window database datastore

Option	Description
<i>Database Type</i>	Select <i>Microsoft</i> .
<i>Database subtype</i>	Select the database subtype: • <i>Azure PaaS</i> • <i>Azure VM</i> • <i>SQL Server on Premise</i> • <i>Azure DW</i> • <i>APS</i> Select <i>SQL Server on Premise</i> to enable the following options: • <i>Change Data Capture (CDC)</i> • <i>Use SSL encryption</i> For <i>Azure DW</i> and <i>APS</i>, you cannot use a create table, truncate table, or drop table statement in a preload or postload statement.
<i>Database Version</i>	Specifies the applicable Microsoft SQL Server version. The <i>Database Subtype</i> you select determines the available versions. For example, select <i>SQL Server on Premise</i> for the subtype, and all Microsoft SQL Server versions are available. However, <i>APS</i> is available in Microsoft SQL Server versions 2016 and 2017.

Option	Description						
CDC dropdown list	Specifies a change data capture (CDC) option. Applicable for <i>SQL Server on Premise</i> only. Select one of the following options from the dropdown list: • <i>No CDC</i>: Disables change data capture. • <i>Native CDC</i>: Uses Microsoft SQL Server change data capture. • <i>Replication Server CDC</i>: Uses the Replication Server change data capture. i Note <i>Replication Server CDC</i> is not available for Microsoft SQL Server 2016 and 2017. If you choose <i>Replication Server CDC</i> and you enable SSL encryption, the SSL encryption applies only to the datastore connection and not to the replication server connection.						
CDC method dropdown list	Specifies the Microsoft SQL Server change data capture method to use. Applicable when you select <i>Native CDC</i> from the CDC dropdown list. Available options are based on the version of Microsoft SQL Server you select. SQL Server version and CDC methods <table><tr><th>2000–2005</th><th>2008</th><th>2012–2017</th></tr><tr><td><i>Replication</i></td><td> • <i>Replication</i> • <i>CDC</i> • <i>Change tracking</i> </td><td> • <i>CDC</i> • <i>Change tracking</i> </td></tr></table> For information about the change data tracking methods, see Use CDC with Microsoft SQL Server databases [page 914].	2000–2005	2008	2012–2017	<i>Replication</i>	• <i>Replication</i> • <i>CDC</i> • <i>Change tracking</i>	• <i>CDC</i> • <i>Change tracking</i>
2000–2005	2008	2012–2017					
<i>Replication</i>	• <i>Replication</i> • <i>CDC</i> • <i>Change tracking</i>	• <i>CDC</i> • <i>Change tracking</i>					
The following options are specific to Microsoft SQL Server when you click <i>Advanced</i> .							

Option	Description
Use Windows Authentication	Specifies whether to use Windows authentication when the datastore accesses the database. No: Uses Microsoft SQL Server authentication to connect to this datastore. No is the default setting. Yes: Uses Windows authentication to connect to this datastore. For more information on how to use Windows authentication with Microsoft SQL Server, refer to the Microsoft SQL Server documentation. i Note Windows authentication is not supported on UNIX.
Use SSL encryption	Specifies whether Data Services connects to the database using SSL/TLS protocol. Applicable only when you choose SQL Server on Premise for the Database Sub-Type. No: Does not connect using SSL protocol. No is the default setting. Yes: Connects using SSL protocol. Enables the Encryption parameters option to open the Encryption Parameters dialog box. i Note SSL encryption slows job performance. If you choose the Replication Server CDC option and Use SSL encryption, SSL encryption applies only to the datastore connection and not to the replication server connection. If the Microsoft SQL Server is installed on a different machine than Data Services, configure an SSL certificate on both the server and the client machine. For more information about administrator tasks for enabling SSL encryption, see the <i>Administrator Guide</i>.

Option	Description
Encryption parameters	Opens the Encryption Parameters dialog box. To open the Encryption Parameters dialog box, either double-click in the empty cell or click the ... icon that appears at the right end of the cell when you click in the cell. To enable the Encryption Parameters option, select Yes for Use SSL encryption. Windows: Open the Encryption Parameters dialog box to enable or disable the Validate Certificate option. Unix/Linux: Open the Encryption Parameters dialog box to enable or disable the Validate Certificate option and to set the Unix-specific encryption parameters when you enable Validate Certificate.
The following options appear in the Encryption Parameters dialog box.	
Validate Certificate	Specifies whether Data Services validates the SSL server certificate. The option is selected by default. • Selected: Data Services validates the SSL server certificate. • Deselected: Data Services accepts any server certificate. It does not validate the server certificate.
UNIX Certificate host	Optional. Unix and Linux only. Specifies the certificate host name, if applicable.
UNIX Trust Store	Unix and Linux only. Specifies the location and file name of the trust store. You must have a trust store on the client machine. The trust store contains one or more SSL certificates, including the certificate that you prepared for the connection. If you leave the option blank, Data Services looks in the default trust store file, <code>DS_MSSQL_cert.pem</code>, to verify the server SSL certificate. The default trust store location is < \$LINK_DIR>/ssl/dbdatastore_certs/.
UNIX Trust Store password	Optional. Unix and Linux only. Specifies the password associated with the specified trust store. If you do not password-protect the specified trust store, leave this option blank.
The following options are available in the General section.	

Option	Description
<i>Rows per commit</i>	Specifies the maximum number of rows loaded to a target table before saving the data. The default value is 1000. This value is the default commit size for target tables in this datastore. You can overwrite this value for individual target tables.
<i>Bulk loader directory</i>	Specifies the location where command and data files are written for bulk loading. Optionally use a variable for this option. i Note Bulk loading is available when you select <i>No CDC</i>. For information about bulk loading in Microsoft SQL Server, see the <i>Performance Optimization Guide</i>.
<i>Overflow file directory</i>	Specifies the location of overflow files written by target tables in this datastore. Optionally use a variable for this option.
The following Changed Data Capture options appear when you configure CDC with the SAP Replication Server.	
CDC with SAP Replication Server	Specifies values for the Replication server. Applicable when you select <i>Replication Server CDC</i> from the CDC dropdown list. Applicable for Microsoft SQL Server versions 2014 and earlier. For information about using the Replication Server for change data capture, see Use CDC with SAP Replication Server [page 898].
<i>Replication Server</i>	Specifies the Replication Server name. The name must match what is defined in the interface file <code>sql.ini</code>. In addition, the name must match the name of the Replication Server to which the Replication Agent is connected.
<i>Replication Server user</i>	Specifies the qualified user of the Replication Server. The user must have create object, primary subscribe, and connect target permission rights.
<i>Replication Server password</i>	Specifies the password for the specified <i>Replication Server user</i>.

Option	Description
<i>Retention Period</i>	Specifies the length of time in minutes that the replication server retains internal properties. Enter the value that corresponds to the Replication Server connection property save_interval property. ❖ Example If your target database back-up interval is 24 hours, set the Retention Period to 24x60=1440 to set the equivalent number of minutes to retain the last 24 hours of changed data.
<i>Replication Agent</i>	Specifies the Replication Agent name, which must match what is defined in the interface file <code>sql.ini</code> .
<i>Replication Agent user</i>	Specifies the Replication Agent user, who must be an "sa" user.
<i>Replication Agent password</i>	Specifies the password related to the specified <i>Replication Agent user</i> .
The following Changed Data Capture options appear when you configure CDC with <i>Native CDC</i> and <i>Replication</i> .	
<i>Distribution server name</i> <i>Distribution database name</i> <i>Publication name</i> <i>Distribution user name</i> <i>Distribution password</i>	Specifies values for the Distribution server that you configure for the Replication Server method of CDC. To select the Replication Server method, Select <i>Native CDC</i> from the CDC dropdown list, and select <i>Replication</i> from the dropdown list under the CDC dropdown list. Applicable for Microsoft SQL Server 2008 and earlier versions. Obtain values from the distribution options generated when you configure the distribution database in Microsoft SQL Enterprise Manager. For information about the Microsoft SQL Server replication method, see Replication Server method [page 926]

Parent topic: [Microsoft SQL Server datastore \[page 129\]](#)

Related Information

[Importing SSL certificate from Microsoft SQL Server \[page 136\]](#)

[Common datastore options \[page 99\]](#)

[Bulk loading in Microsoft SQL Server](#)

9.1.8.2 Importing SSL certificate from Microsoft SQL Server

If you plan to connect to SQL Server using SSL encryption and validating the SSL certificate, you must first import the SSL certificate from Microsoft SQL Server.

1. Generate a private key file and get a server certificate for Microsoft SQL Server from a trusted authority.
As an option, you can create your own private key file and server certificate. For information about SSL in Data Services, see the *Administrator Guide*. Also see your Microsoft SQL Server documentation.
2. Open Microsoft Management Console (MMC).
For information about MMC, see the Microsoft Web site <https://msdn.microsoft.com/en-us/library/bb742442.aspx> .
3. Import the Microsoft SQL Server database certificate file to the Trusted Root Certification Authorities folder of the current computer or the current user.
The import places the SSL certificate in your TrustStore.
4. Close MMC.
5. Open SQL Server client and enter the connection information.
6. Select *Use Encryption*. If the connection fails, check your SSL configuration.

Task overview: [Microsoft SQL Server datastore \[page 129\]](#)

Related Information

[Microsoft SQL Server database datastore options \[page 130\]](#)

9.1.9 Memory datastores

A memory datastore is a container for memory tables.

Memory tables are schemas that allow you to cache intermediate data. Memory tables can cache data from relational database tables and hierarchical data files. Hierarchical data files can contain nested schemas, such as XML messages and SAP IDocs.

Memory datastores enhance processing performance for data flows in real-time jobs that consist of small amounts of data. Data Services stores the data in memory to provide immediate access instead of accessing the original source data.

A memory datastore contains memory table schemas saved in the repository. In contrast, a regular database datastore provides a connection to a database, application, or adapter.

Memory tables keep data in memory only for the duration of the job. The following table describes the advantages of using memory tables related to data storage.

Use	Advantages
To move data between data flows in real-time jobs	Memory tables improve job performance when the job has multiple data flows. Memory tables cache intermediate data. Data flows access intermediate data from cache instead of from the remote database. For best performance, use memory tables only when you process small quantities of data.
To store table data in memory for the duration of a job	Memory tables improve function performance in transforms. Memory tables store table data in memory. Functions that don't require database operations, such as the Lookup_Ext function, access data from memory without having to read it from a remote database.

Memory table restrictions:

- You cannot share data in memory tables between different real-time jobs.
- You cannot use memory tables for batch jobs.

i Note

For batch jobs, consider using persistent cache datastores that contain persistent cache tables.

[Defining a memory datastore \[page 138\]](#)

Create and define a new memory datastore in the same manner as creating a database datastore.

[Memory datastore option descriptions \[page 139\]](#)

Create a Memory database datastore for use in Data Services.

[Creating a memory table \[page 140\]](#)

Create a memory table and save it with the associated memory datastore.

[Using a memory table as a source or target \[page 140\]](#)

Use a memory table as a source or target in a real-time job.

[Updating a target memory table schema \[page 141\]](#)

When schemas change in upstream objects in a real-time data flow, you can quickly update the schema for the target memory table.

[Memory table target options \[page 142\]](#)

Target options for a memory table are options that are common to most target tables.

[Use Row ID to enhance expression performance \[page 142\]](#)

Select to add a Row ID to a memory table to enable certain functions to more quickly iterate through data to run expressions.

[Troubleshooting memory tables \[page 144\]](#)

You may experience a few problems when you use memory tables.

Parent topic: [Database datastores \[page 93\]](#)

Related Information

[Defining a database datastore \[page 96\]](#)
[Datastore metadata \[page 102\]](#)
[Imported metadata from database datastores \[page 107\]](#)
[Attunity Connector \(mainframe datastore\) \[page 115\]](#)
[Data Federator \[page 121\]](#)
[IBM DB2 datastores \[page 124\]](#)
[Informix \[page 127\]](#)
[Microsoft SQL Server datastore \[page 129\]](#)
[MySQL datastore \[page 145\]](#)
[ODBC \[page 161\]](#)
[Oracle datastores \[page 166\]](#)
[Netezza \[page 170\]](#)
[SAP SQL Anywhere \[page 172\]](#)
[SAP ASE \[page 173\]](#)
[Sybase IQ \[page 174\]](#)
[Teradata \[page 179\]](#)

9.1.9.1 Defining a memory datastore

Create and define a new memory datastore in the same manner as creating a database datastore.

1. Select **Project** > **New** > **Datastore**.

The datastore editor opens.

2. Type a name for the datastore in the **Name** text box.

If you have adopted Data Services naming conventions, ensure that you prefix the datastore name with **Memory_DS_**. The software appends datastore names to table names when you add a memory table to a data flow in the workspace. The software uses regular table icons to represent memory tables. Therefore, label a memory datastore to distinguish the memory tables from regular database tables in the workspace.

3. Keep the default setting of **Database** for **Datastore type**.
4. Select **Memory** from the **Database Type** dropdown list.

There are no additional attributes or advanced options to add for a memory datastore.

5. Click **OK**.

The software creates the memory datastore. The new datastore appears alphabetically in the **Datastore** tab of the object library.

Task overview: [Memory datastores \[page 136\]](#)

Related Information

[Memory datastore option descriptions \[page 139\]](#)

[Creating a memory table \[page 140\]](#)

[Using a memory table as a source or target \[page 140\]](#)

[Updating a target memory table schema \[page 141\]](#)

[Memory table target options \[page 142\]](#)

[Use Row ID to enhance expression performance \[page 142\]](#)

[Troubleshooting memory tables \[page 144\]](#)

9.1.9.2 Memory datastore option descriptions

Create a Memory database datastore for use in Data Services.

The following table contains configuration descriptions specific for the Memory datastore.

Bulk Loader

Option	Description
<i>JS and DB on same machine</i>	<i>Yes:</i> The Job Server and the database are on the same computer. <i>No:</i> The Job Server and the database are on separate computers. This is the default setting.

Parent topic: [Memory datastores \[page 136\]](#)

Related Information

[Defining a memory datastore \[page 138\]](#)

[Creating a memory table \[page 140\]](#)

[Using a memory table as a source or target \[page 140\]](#)

[Updating a target memory table schema \[page 141\]](#)

[Memory table target options \[page 142\]](#)

[Use Row ID to enhance expression performance \[page 142\]](#)

[Troubleshooting memory tables \[page 144\]](#)

[Locales and Multi-byte Functionality](#)

[Common datastore options \[page 99\]](#)

9.1.9.3 Creating a memory table

Create a memory table and save it with the associated memory datastore.

Open an existing real-time job or begin to create a new real-time job in the workspace.

1. Click the template table icon  in the tool palette and click in the workspace.

The *Create Template* dialog box opens.

2. Enter a table name in the *Template name* text box.
3. Select the applicable memory datastore from the *In datastore* dropdown list.
Data Services replaces the *Owner name* text box with the *Create row ID* checkbox.
4. Optional. To include a system-generated row ID column in the table, check *Create Row ID*.
5. Click *OK*.

The memory table appears in the workspace as a template table icon.

6. Connect the memory table to the data flow as a target.
7. From the *Project*, menu select *Save*.

The template table icon changes to a target table icon.

The new memory table appears under the Table node for the applicable memory datastore in the *Datastore* tab of the object library

After you use the memory table as a target in a data flow, you can also use it as a source in a data flow.

Task overview: [Memory datastores \[page 136\]](#)

Related Information

[Defining a memory datastore \[page 138\]](#)

[Memory datastore option descriptions \[page 139\]](#)

[Using a memory table as a source or target \[page 140\]](#)

[Updating a target memory table schema \[page 141\]](#)

[Memory table target options \[page 142\]](#)

[Use Row ID to enhance expression performance \[page 142\]](#)

[Troubleshooting memory tables \[page 144\]](#)

9.1.9.4 Using a memory table as a source or target

Use a memory table as a source or target in a real-time job.

A memory table is not available in the object library until you add a template table to a real-time data flow, configure it as a memory table, and connect it as a target.

Open an applicable real-time data flow in the workspace.

1. Open the [Datastores](#) tab in the object library.
2. Expand the applicable memory datastore node and the table node.

Data Services lists the memory tables that belong to the memory datastore.
3. Drag the applicable memory table from the object library and drop it into position in the real-time data flow in the workspace.

The software displays a menu.
4. Select [Make Source](#) or [Make Target](#) as applicable from the menu.
5. Connect the memory table as a source or target in the data flow.
6. Continue configuring the data flow as usual.
7. Save the job.

Task overview: [Memory datastores \[page 136\]](#)

Related Information

[Defining a memory datastore \[page 138\]](#)

[Memory datastore option descriptions \[page 139\]](#)

[Creating a memory table \[page 140\]](#)

[Updating a target memory table schema \[page 141\]](#)

[Memory table target options \[page 142\]](#)

[Use Row ID to enhance expression performance \[page 142\]](#)

[Troubleshooting memory tables \[page 144\]](#)

9.1.9.5 Updating a target memory table schema

When schemas change in upstream objects in a real-time data flow, you can quickly update the schema for the target memory table.

Perform the steps under the following circumstance:

- You have adjusted the schema of upstream objects in a real-time data flow.
 - Your target is a memory table.
 - You perform [View where Used](#) on the memory table. The changes you make to the schema here change the memory table schema everywhere it is used. If more than one object calls the memory table, consider creating a new memory table instead of updating the schema.
1. Right-click the icon for the memory table target object in the workspace.
 2. Select [Update Schema](#) from the dropdown menu.

The software uses the schema of the preceding object to update the memory table schema.

Task overview: [Memory datastores \[page 136\]](#)

Related Information

[Defining a memory datastore \[page 138\]](#)
[Memory datastore option descriptions \[page 139\]](#)
[Creating a memory table \[page 140\]](#)
[Using a memory table as a source or target \[page 140\]](#)
[Memory table target options \[page 142\]](#)
[Use Row ID to enhance expression performance \[page 142\]](#)
[Troubleshooting memory tables \[page 144\]](#)

9.1.9.6 Memory table target options

Target options for a memory table are options that are common to most target tables.

The target editor contains a *Schema In* pane and a *Schema Out* pane in the upper portion of the editor. The lower pane contains option tabs.

For descriptions of the common options in target tables, see the *Reference Guide*.

Parent topic: [Memory datastores \[page 136\]](#)

Related Information

[Defining a memory datastore \[page 138\]](#)
[Memory datastore option descriptions \[page 139\]](#)
[Creating a memory table \[page 140\]](#)
[Using a memory table as a source or target \[page 140\]](#)
[Updating a target memory table schema \[page 141\]](#)
[Use Row ID to enhance expression performance \[page 142\]](#)
[Troubleshooting memory tables \[page 144\]](#)

9.1.9.7 Use Row ID to enhance expression performance

Select to add a Row ID to a memory table to enable certain functions to more quickly iterate through data to run expressions.

The *Create Row ID* option is available in the *Create Table* dialog box. The software displays the *Create Table* dialog box when you create a memory table from a template table in a data flow.

When you select *Create Row ID*, the software generates an integer column called `DI_Row_ID` in the target table. The software assigns a value of 1 in the `DI_Row_ID` column for the first row inserted in the target table. The

software continues assigning values for each added row by adding 1 to the previous assigned value. Use the `DI_Row_ID` column in expressions to iterate through a table using a `lookup_ext` function in a script.

Note

The same functionality is available for other datastore types using the SQL function.

❖ Example

```
$NumOfRows = total_rows (memory_DS..table1)
$I = 1;
$count=0
while ($count < $NumOfRows)
begin
    $data =
        lookup_ext([memory_DS..table1, 'NO_CACHE', 'MAX'], [A], [O], [DI_Row_ID, '=',
$I]);
    $I = $I + 1;
    if ($data != NULL)
    begin
        $count = $count + 1;
    end
end
```

In the example script, `table1` is a memory table. Because memory tables do not have table owners, the table name contains a dot, a space, and a dot. The space is where a table owner name would go if the table wasn't a memory table. In the example script, the memory table uses the Data Services naming convention that contains the prefix `memory_DS_` in the table name.

If you use the function editor to create the `lookup_ext` function, use the applicable function arguments in the editor. The 7th line in the example contains the function argument values `NO_CACHE` and `MAX`.

The first line of the example contains the `total_rows` function: `total_rows (memory_ds..table1)`. The `total_rows` function returns the number of rows in a particular table in a datastore. You can use this function with any type of datastore.

The software also provides a built-in function named `truncate_table` that is not shown in the example. Use `truncate_table` to explicitly expunge data from a memory table. The `truncate_table` function provides finer control than the active job has over your data and memory usage. Use the `truncate_table` function only with memory tables. The syntax is: `truncate_table( <DatastoreName..TableName> )`.

For complete descriptions of Data Services built-in functions, see the *Reference Guide*.

Parent topic: [Memory datastores \[page 136\]](#)

Related Information

[Defining a memory datastore \[page 138\]](#)

[Memory datastore option descriptions \[page 139\]](#)

[Creating a memory table \[page 140\]](#)

[Using a memory table as a source or target \[page 140\]](#)

[Updating a target memory table schema \[page 141\]](#)

[Memory table target options \[page 142\]](#)
[Troubleshooting memory tables \[page 144\]](#)
[lookup_ext](#)
[truncate_table](#)

9.1.9.8 Troubleshooting memory tables

You may experience a few problems when you use memory tables.

When you use memory datastores and memory tables, there are a few problems that can occur when you execute the job. The following lists some potential problems and their solutions.

Problem	Solution
Job execution stops before the job is complete.	Check virtual memory space. When the software runs out of virtual memory, it stops the job. Memory tables use virtual memory to store data while processing. Caching data in virtual memory is a feature that increases performance.
Job execution stops because of a runtime error.	Ensure that the target schema matches the schema of the upstream object in the data flow. Consider using the Update Schema feature to ensure that the target schema matches the upstream schema.
Other runtime errors occur that prevent the job from completing	Examine the following 2 log files that contain information specific to memory tables: • <code>trace_memory_reader</code> • <code>trace_memory_loader</code> For more information about debugging using log files, see Access logs [page 761].

Parent topic: [Memory datastores \[page 136\]](#)

Related Information

[Defining a memory datastore \[page 138\]](#)
[Memory datastore option descriptions \[page 139\]](#)
[Creating a memory table \[page 140\]](#)
[Using a memory table as a source or target \[page 140\]](#)
[Updating a target memory table schema \[page 141\]](#)
[Memory table target options \[page 142\]](#)

[Use Row ID to enhance expression performance \[page 142\]](#)

9.1.10 MySQL datastore

To use MySQL data in your SAP Data Services processes, create a MySQL database datastore to access your MySQL data.

Before you create a MySQL datastore, ensure that you perform the required prerequisite tasks. First, you must use a supported MySQL connector/ODBC driver. For Linux, Solaris, and AIX, you must also have the unixODBC driver manager. Read about the unixODBC driver manager in your MySQL documentation.

Data Services supports either a DSN or a server-based (DSN-less) connection in MySQL datastores. Optionally, include FTP protocol, and, if you have the supported versions, include SSL/TLS encryption for Windows, Linux, and Solaris.

[MySQL datastore options \[page 146\]](#)

To create a MySQL datastore, complete applicable options in the datastore editor.

[Configuring ODBC driver on Windows \[page 149\]](#)

For Windows, download and install the applicable version of the MySQL ODBC driver, and verify that the driver is installed using the SAP Data Services ODBC Drivers Selector.

[MySQL dependency on unixODBC Driver Manager \[page 149\]](#)

MySQL on Linux, Solaris, and AIX has a dependency on the unixODBC Driver Manager.

[Configuring MySQL ODBC driver for Linux, Solaris, and AIX \[page 150\]](#)

For Linux, Solaris, and AIX, install the MySQL ODBC connector/driver, and configure the driver using the SAP Data Services Connection Manager (Connection Manager).

[SSL/TLS encryption for MySQL \[page 154\]](#)

When you use a datastore to access your MySQL data, you can optionally include SSL/TLS encryption.

Parent topic: [Database datastores \[page 93\]](#)

Related Information

[Defining a database datastore \[page 96\]](#)

[Datastore metadata \[page 102\]](#)

[Imported metadata from database datastores \[page 107\]](#)

[Attunity Connector \(mainframe datastore\) \[page 115\]](#)

[Data Federator \[page 121\]](#)

[IBM DB2 datastores \[page 124\]](#)

[Informix \[page 127\]](#)

[Microsoft SQL Server datastore \[page 129\]](#)

[Memory datastores \[page 136\]](#)

[ODBC \[page 161\]](#)

[Oracle datastores \[page 166\]](#)

[Netezza \[page 170\]](#)
[SAP SQL Anywhere \[page 172\]](#)
[SAP ASE \[page 173\]](#)
[Sybase IQ \[page 174\]](#)
[Teradata \[page 179\]](#)
[Defining a database datastore \[page 96\]](#)

9.1.10.1 MySQL datastore options

To create a MySQL datastore, complete applicable options in the datastore editor.

The following table contains descriptions for options specific to the MySQL datastore.

Option descriptions for MySQL

Option	Description
<i>Database Version</i>	Specifies the database version. Select the version you are using for this datastore. i Note Support for DSN-less connection with SSL/TLS protocol is applicable for MySQL Client version 5.3.X and SAP Data Services 4.2.12 and later versions.
<i>Use data source name (DSN)</i>	Specifies whether to use a DSN connection. If you select <i>Use data source name (DSN)</i>, the <i>ODBC Admin...</i> button appears. Either create the DSN before you create the datastore, or open the ODBC Data Source Administrator by clicking <i>ODBC Admin...</i> For instructions, see Windows: Configuring DSN and SSL/TLS encryption for MySQL datastores [page 155]. To use a server name (DSN-less) connection, do not select <i>Use data source name (DSN)</i>.
Complete the following options to configure a server-name (DSN-less) connection:	
<i>Database server name</i> <i>Database name</i> <i>Port</i> <i>User Name</i> <i>Password</i>	Specifies the server on which the database resides, the database to connect to, and the user name and password to access the database.
The following options appear when you select <i>Use data source name (DSN)</i>:	

Option	Description
<i>Data source name</i>	Select the DSN you created for this database from the drop-down list. If you plan to use the auto correct load feature for MySQL targets, ensure that you created the DSN with a user who has permission to create or replace stored procedures.

The following table contains descriptions for options that appear when you click [Advanced](#).

Advanced datastore option descriptions for MySQL

Option	Description
<i>Use SSL encryption</i>	Specifies whether to use SSL/TLS protocol to encrypt SSL/TLS certificate and server authentication. YES: Connects to the database using SSL/TLS protocol. NO: Connects to the database without using SSL/TLS protocol. This option is available only when you create a DSN-less connection.
<i>Encryption parameters</i>	Opens the Encryption Parameters dialog box. Either click the option cell and click the "..." to open the dialog box, or double-click in the cell. For option descriptions see MySQL SSL encryption parameters for DSN-less connections [page 158].
<i>Database Server working directory</i>	Specifies a working directory for the load utility. Enter or browse to the location of the working directory. The working directory for the load utility is on the computer that runs the MySQL server. Complete this option whenever the MySQL server and the Job Server run on separate machines.
<i>FTP host name</i>	Specifies the FTP host name. Use the computer name, fully qualified domain name, or IP address. If you leave this field blank or enter the name of the host computer for the Job Server, Data Services assumes that MySQL shares the same computer and that FTP is unnecessary. When FTP is unnecessary, Data Services does not require values for the remaining FTP fields
<i>FTP login user name</i>	Specifies the user name for the person authorized to access FTP.
<i>FTP login password</i>	Specifies the password to log into the FTP.
<i>FTP host working directory</i>	Specifies the absolute path for the database server where Data Services transfers the data file between the Job Server and the MySQL server. For Windows servers, configure a path to a virtual directory.
ODBC miscellaneous options	

Option	Description
<i>Date format</i>	Specifies a date format that the ODBC driver and the data source support. The default date format is <code>yyyy-mm-dd</code> .
<i>Time format</i>	Specifies a time format that the ODBC driver and the data source support. The default time format is <code>hh:mi:ss</code> .
<i>Date-time format</i>	Specifies a date-time format that the ODBC driver and the database support. The default date-time format is <code>yyyy-mm-dd hh:mi:ss</code> .
<i>Decimal separator</i>	Specifies a character to separate the decimal portion of a number. The default character is a period.
<i>Data type conversion support</i>	Specifies the action to take when there is a data type mismatch in an expression. AUTOMATIC: Generates an explicit convert function call automatically. AUTOMATIC is the default setting. ODBC SYNTAX: Generates an explicit convert function call using ODBC syntax. SQL-92 SYNTAX: Generates an explicit convert function call using SQL-92 syntax. NO: Does not generate an explicit convert function call when there is a data type mismatch in an expression.
<i>NVL support</i>	Specifies that when the input value is NULL, replace with a specified value. AUTOMATIC: The default setting. ODBC SYNTAX: Use ODBC syntax to replace input NULL values with a specified value. NO: Does not replace input values of NULL with a specified value.
<i>Ifthenelse support</i>	Specifies whether to use conditional logic in mapping and selecting operations. NO: Does not allow conditional logic in mapping and selection operations. NO is the default. YES: Allows conditional logic in mapping and selection operations.

Parent topic: [MySQL datastore \[page 145\]](#)

Related Information

[Configuring ODBC driver on Windows \[page 149\]](#)

[MySQL dependency on unixODBC Driver Manager \[page 149\]](#)

[Configuring MySQL ODBC driver for Linux, Solaris, and AIX \[page 150\]](#)

[SSL/TLS encryption for MySQL \[page 154\]](#)

[Common datastore options \[page 99\]](#)

9.1.10.2 Configuring ODBC driver on Windows

For Windows, download and install the applicable version of the MySQL ODBC driver, and verify that the driver is installed using the SAP Data Services ODBC Drivers Selector.

1. Open the ODBC Drivers Selector in `<LINK_DIR>\bin\ODBCDriversSelector.exe`.
2. Select the ODBC driver for MySQL under the *Database version* column.
3. Click the corresponding cell under the *ODBC Drivers* column and select the correct driver from the dropdown list.
4. Look at the value in the *State* column. If the status doesn't appear in the *State* column, click in the empty cell and the applicable status appears.

The state should be "Installed."

If the status is anything other than "Installed," you may not have properly installed the MySQL ODBC driver.

Task overview: [MySQL datastore \[page 145\]](#)

Related Information

[MySQL datastore options \[page 146\]](#)

[MySQL dependency on unixODBC Driver Manager \[page 149\]](#)

[Configuring MySQL ODBC driver for Linux, Solaris, and AIX \[page 150\]](#)

[SSL/TLS encryption for MySQL \[page 154\]](#)

9.1.10.3 MySQL dependency on unixODBC Driver Manager

MySQL on Linux, Solaris, and AIX has a dependency on the unixODBC Driver Manager.

The unixODBC Driver Manager handles the calls from MySQL, and passes applicable calls to the MySQL Connector/ODBC driver.

If you do not have the unixODBC Driver Manager on your system, download it from the UNIX ODBC organization Web site at <http://www.unixodbc.org> .

Consult your *MySQL Connector/ODBC Developer Guide* to learn about the Connector/ODBC architecture and how the driver uses the driver manager to access your database.

Make sure that when you install the driver manager, you enter the directory location of the unixODBC Driver Manager `libodbc.so` as the first directory in the beginning of the string `LD_LIBRARY_PATH/LIBPATH/SHLIB_PATH`.

To display the `LD_LIBRARY_PATH`, use the following command:

```
echo $LD_LIBRARY_PATH
```

⚠ Caution

If the first directory in `LD_LIBRARY_PATH/LIBPATH/SHLIB_PATH` is not the same as where you installed `libodbc.so`, a job that uses MySQL as a source, target, or a repository may not work as expected.

Parent topic: [MySQL datastore \[page 145\]](#)

Related Information

[MySQL datastore options \[page 146\]](#)

[Configuring ODBC driver on Windows \[page 149\]](#)

[Configuring MySQL ODBC driver for Linux, Solaris, and AIX \[page 150\]](#)

[SSL/TLS encryption for MySQL \[page 154\]](#)

9.1.10.4 Configuring MySQL ODBC driver for Linux, Solaris, and AIX

For Linux, Solaris, and AIX, install the MySQL ODBC connector/driver, and configure the driver using the SAP Data Services Connection Manager (Connection Manager).

1. Install the unixODBC Driver Manager. For information, see [MySQL dependency on unixODBC Driver Manager \[page 149\]](#).
2. Follow the steps for your specific platform to download and install the MySQL Connector/ODBC driver.
3. Follow the process for configuring ODBC data sources using the Connection Manager. For instructions, see [Linux/Solaris: Configuring DSN and SSL/TLS encryption for MySQL datastores \[page 157\]](#).

[Installing the Connector/ODBC driver for Linux \[page 151\]](#)

After you obtain the applicable Connector/ODBC driver file for Linux, extract and install it on your system.

[Installing the Connector/ODBC driver for Solaris \[page 151\]](#)

After you obtain the applicable Connector/ODBC driver for Solaris, extract and install it on your system.

[Installing the Connector/ODBC driver for AIX \[page 153\]](#)

After you obtain the applicable Connector/ODBC driver file for AIX, extract and install it on your system.

Task overview: [MySQL datastore \[page 145\]](#)

Related Information

[MySQL datastore options \[page 146\]](#)

[Configuring ODBC driver on Windows \[page 149\]](#)

[MySQL dependency on unixODBC Driver Manager \[page 149\]](#)

[SSL/TLS encryption for MySQL \[page 154\]](#)

9.1.10.4.1 Installing the Connector/ODBC driver for Linux

After you obtain the applicable Connector/ODBC driver file for Linux, extract and install it on your system.

Ensure that use a version of Linux that SAP Data Services supports. See the [Product Availability Matrix \(PAM\)](#) on the SAP Support Portal for an up-to-date listing of supported versions.

Select one of the following methods to install the Connector/ODBC driver libraries on Linux:

1. Install the bundled RPM unixODBC package on the Linux installation CD. For example, the RPM package name on Redhat 5, 64-bit is `unixODBC-2.2.11-7.1.rpm`.
2. Download and install the 64-bit unixODBC (x86_64) package from the following location: <http://sourceforge.net/projects/unixodbc/files/unixODBC/2.2.14/unixODBC-2.2.14-linux-x86-64.tar.gz/download> . For example, the file you download is `unixODBC-2.2.14-Linux-x86-64.tar.gz`.

Task overview: [Configuring MySQL ODBC driver for Linux, Solaris, and AIX \[page 150\]](#)

Related Information

[Installing the Connector/ODBC driver for Solaris \[page 151\]](#)

[Installing the Connector/ODBC driver for AIX \[page 153\]](#)

9.1.10.4.2 Installing the Connector/ODBC driver for Solaris

After you obtain the applicable Connector/ODBC driver for Solaris, extract and install it on your system.

Ensure that use a version of Solaris that SAP Data Services supports. See the [Product Availability Matrix \(PAM\)](#) on the SAP Support Portal for an up-to-date listing of supported versions. In addition:

- Ensure that have the Sun C++ compiler version 5.5 or higher.
 - Ensure that you download the unixODBC source.
1. Download and extract the unixODBC package.
 - a. Download `unixODBC-<version>.tar.gz` from <http://www.unixodbc.org>  to the `$TEMP` directory.
 - b. Extract the package with `gunzip` and `tar`.

```
gunzip unixODBC-<version>.tar.gz
tar -xvf unixODBC-<version>.tar
```

- c. Change to the newly created `unixODBC-<version>` directory.

```
cd $TEMP/unixODBC-<version>
```

2. Make the libraries and programs.

- a. Ensure that CC (the Sun C++ compiler) is in the `PATH` environment variable, and add it if necessary.

- o Using ksh:

```
export PATH=/home4/thirdparty/software/sunonecc/8.0-sj/SUNWspro/bin/CC:
$PATH
```

- o Using csh:

```
setenv PATH /home4/thirdparty/software/sunonecc/8.0-sj/SUNWspro/bin/CC:
$PATH
```

- b. Build the package using the standard GNU autoconf process.

```
./configure CFLAGS="-xarch=v9" LDFLAGS="-xarch=v9" CXXFLAGS="-xarch=v9" --
enable-gui=no
make
make install
```

i Note

By default, the files are installed to `/usr/local`. You can specify a different location by altering the prefix option:

```
./configure --prefix=<new_location>/unixODBC CFLAGS="-xarch=v9"
LDFLAGS="-xarch=v9" CXXFLAGS="-xarch=v9" --enable-gui=no
```

where `<new_location>` is the location where you want to install the unixODBC libraries.

3. Add the unixODBC library to the `$LD_LIBRARY_PATH` environment variable.

For example:

```
export LD_LIBRARY_PATH=<install_path>/unixODBC/lib:$LD_LIBRARY_PATH
```

where `<install_path>` is the location where all the unixODBC libraries are installed.

Task overview: [Configuring MySQL ODBC driver for Linux, Solaris, and AIX \[page 150\]](#)

Related Information

[Installing the Connector/ODBC driver for Linux \[page 151\]](#)

[Installing the Connector/ODBC driver for AIX \[page 153\]](#)

9.1.10.4.3 Installing the Connector/ODBC driver for AIX

After you obtain the applicable Connector/ODBC driver file for AIX, extract and install it on your system.

Ensure that use a version of Linux that SAP Data Services supports. See the [Product Availability Matrix \(PAM\)](#) on the SAP Support Portal for an up-to-date listing of supported versions. In addition:

- Ensure that you have the VisualAge C++ compiler version 6.0 or higher.
 - Ensure that you download the unixODBC source.
1. Download and extract the unixODBC package.
 - a. Download `unixODBC-<version>.tar.gz` from <http://www.unixodbc.org>  to the `$TEMP` directory.
 - b. Extract the package with `gunzip` and `tar`.

```
gunzip unixODBC-<version>.tar.gz
tar -xvf unixODBC-<version>.tar
```

- c. Change to the newly created `unixODBC-<version>` directory.

```
cd $TEMP/unixODBC-<version>
```

2. Make the libraries and programs.
 - a. Ensure that `xlc` (the C++ compiler) is in the `PATH` environment variable, and add it if necessary.
 - Using `ksh`:

```
export PATH=/usr/vacpp/bin:$PATH
```

- Using `csh`:

```
setenv PATH /usr/vacpp/bin:$PATH
```

- b. Configure the C compiler to be thread-enabled:

```
export CC=xlc_r
export CCC=xlc_r
```

- c. To compile a 64-bit version of the driver manager using the `xlc_r` compilers, set the `OBJECT_MODE` and `CFLAGS` environment variables:

```
export OBJECT_MODE=64
export CFLAGS=-q64
```

- d. Build the package.

```
./configure --enable-gui=no --enable-drivers=no
make
make install
```

Note

By default, the files are installed to `/usr/local`. You can specify a different location by altering the `prefix` option:

```
./configure --prefix=<new_location> --enable-gui=no --enable-drivers=no
```

where `<new_location>` is the location where you want to install the unixODBC libraries.

- e. If you plan to load the driver manager dynamically from `/prefix/lib`, extract `libodbc.a`, `libodbcinst.a`, and `libodbcrc.a`.

```
ar -x -X 64 libodbc.a
ar -x -X 64 libodbcinst.a
ar -x -X 64 libodbcrc.a
```

- f. Create the dynamically linked library.

```
ln -s libodbcinst.so.1 libodbcinst.so
```

3. Add the unixODBC library to the `$LIBPATH` environment variable.

For example:

```
export LIBPATH=<install_path>:$LIBPATH
```

where `<install_path>` is the location where all the unixODBC libraries are installed.

Task overview: [Configuring MySQL ODBC driver for Linux, Solaris, and AIX \[page 150\]](#)

Related Information

[Installing the Connector/ODBC driver for Linux \[page 151\]](#)

[Installing the Connector/ODBC driver for Solaris \[page 151\]](#)

9.1.10.5 SSL/TLS encryption for MySQL

When you use a datastore to access your MySQL data, you can optionally include SSL/TLS encryption.

SAP Data Services supports SSL/TLS encryption for both DSN or DSN-less connections on Windows, Linux, and Solaris operating systems.

i Note

Support for SSL/TLS begins with Data Services version 4.2.12 for Windows, Linux and Solaris only. Supported MySQL versions for SSL/TLS are:

- MySQL server 5.x
- MySQL client 5.3
- MySQL ODBC Connector 5.3.x

See the [Product Availability Matrix \(PAM\)](#) on the SAP Support Portal for details about supported versions and platforms.

! Restriction

Data Services does not support SSL/TLS for AIX. For MySQL and AIX, the PAM may have incorrect versions listed under the Client column. For AIX, Data Services supports the MySQL ODBC Connector version **5.1.8**, which is not listed in the PAM, and which does not support SSL/TLS.

See your MySQL documentation for instructions to enable SSL/TLS for your MySQL server, and to generate SSL/TLS certificate and key files. Also, for Linux and Solaris platforms, read about the unixODBC driver manager and how it relates to MySQL Connector/ODBC driver.

[Windows: Configuring DSN and SSL/TLS encryption for MySQL datastores \[page 155\]](#)

To configure a DSN connection and configure the optional SSL/TLS protocol for a MySQL datastore, use the SAP Data Services Connection Manager.

[Linux/Solaris: Configuring DSN and SSL/TLS encryption for MySQL datastores \[page 157\]](#)

To configure a DSN connection and configure the optional SSL/TLS protocol for a MySQL datastore, use the SAP Data Services Connection Manager.

[MySQL SSL encryption parameters for DSN-less connections \[page 158\]](#)

Configure SSL/TLS protocol for MySQL DSN-less connections in the datastore editor in Designer.

Parent topic: [MySQL datastore \[page 145\]](#)

Related Information

[MySQL datastore options \[page 146\]](#)

[Configuring ODBC driver on Windows \[page 149\]](#)

[MySQL dependency on unixODBC Driver Manager \[page 149\]](#)

[Configuring MySQL ODBC driver for Linux, Solaris, and AIX \[page 150\]](#)

9.1.10.5.1 Windows: Configuring DSN and SSL/TLS encryption for MySQL datastores

To configure a DSN connection and configure the optional SSL/TLS protocol for a MySQL datastore, use the SAP Data Services Connection Manager.

Before you create a MySQL datastore in SAP Data Services Designer, perform the following prerequisite tasks:

- Download and install the supported ODBC driver.
 - For Windows, configure the ODBC driver by following the steps in [Configuring ODBC driver on Windows \[page 149\]](#).
 - For Linux and Solaris, install the unix ODBC driver by following the steps for the applicable platform under [MySQL dependency on unixODBC Driver Manager \[page 149\]](#).
 - Access the ODBC Data Source Administrator in one of two ways:
 - Create the MySQL database datastore, select *Use data source name (DSN)*, and click *ODBC...*
 - Open the ODBC Data Source Administrator using the Start menu in Windows.
 - If you plan to configure SSL/TLS encryption, follow specific directions in your MySQL documentation to generate the SSL certificate and key files, and to enable SSL on your MySQL server.
1. In the ODBC Data Source Administrator, select either the **User DSN** or **System DSN** tab and click **Add**.

Select the **User DSN** tab to create a DSN that is visible only to you. Select the **System DSN** tab to create a DSN that is available to all users.

2. Select the applicable MySQL ODBC driver from the list of drivers and click [Finish](#).
3. To create the DSN, complete the applicable options in the **Connection Parameters** dialog box.
 - a. Complete the following options:
 - **Data Source Name**
 - **Description** (optional)
 - **TCP/IP Server**: Enter the MySQL server host to access.
 - **Port**: To communicate over a different TCP/IP port than the displayed number, enter a different port number.
 - **User**
 - **Password**
 - **Database**: The dropdown list contains databases for which the specified user has access permission.

i Note

If you plan to use the auto correct load feature for MySQL targets, ensure that you enter a user who has permission to create or replace stored procedures.

- b. Click **OK** to save the DSN.
 - c. Click **Test** to test the connection.
4. Configure SSL/TLS.
 - a. With the ODBC Data Source Administrator still open, click **Details** located at the bottom of the **Connection Parameters** dialog box.
 - b. Select the **SSL** tab.
 - c. Enter values for the SSL/TLS options as applicable.

See SSL/TLS option descriptions in [MySQL SSL encryption parameters for DSN-less connections \[page 158\]](#).
5. Save your settings and close the ODBC Data Source Administrator.

Create a MySQL datastore following the steps in [Defining a database datastore \[page 96\]](#). See option descriptions for MySQL datastores at [MySQL datastore options \[page 146\]](#) and [MySQL SSL encryption parameters for DSN-less connections \[page 158\]](#).

Task overview: [SSL/TLS encryption for MySQL \[page 154\]](#)

Related Information

[Linux/Solaris: Configuring DSN and SSL/TLS encryption for MySQL datastores \[page 157\]](#)
[MySQL SSL encryption parameters for DSN-less connections \[page 158\]](#)

9.1.10.5.2 Linux/Solaris: Configuring DSN and SSL/TLS encryption for MySQL datastores

To configure a DSN connection and configure the optional SSL/TLS protocol for a MySQL datastore, use the SAP Data Services Connection Manager.

Before you perform the steps to configure the DSN connection and SSL/TLS protocol, complete the following prerequisite tasks:

- Install the unix ODBC driver by following the steps for the applicable platform under [MySQL dependency on unixODBC Driver Manager \[page 149\]](#).
- If you plan to configure SSL/TLS encryption, follow specific directions in your MySQL documentation to generate the SSL certificate and key files, and to enable SSL on your MySQL server.
- If you use the RSA cipher in addition to SSL/TLS, create the RSA certificate file following instructions in your MySQL documentation.
- Optional. Create a user interface (UI) for the Connection Manager using the GTK+2 library. Download and install the GTK+2 library from <https://www.gtk.org/> .

1. Open the Connection Manager by entering the following command:

```
$ cd $LINK_DIR/bin/  
$ ./DSConnectionManager.sh
```

2. Open the [Data Sources](#) tab and click [Add](#).

The [Configuration for...](#) dialog box opens with the value filled in for the following options:

- The absolute location of the `odbc.ini` file in which the DSN is defined
- Driver
- Driver version

3. Provide the following values for the driver properties:

- Driver location and file name
- Host name
- Port number
- unixODBC library path

4. Press to go to the main menu.

5. To create the DSN, enter the following values in the [Data Sources](#) tab:

- DSN name
- Database user name
- Database password

6. Press to go to the main menu.

7. To configure the SSL/TLS protocol, select an option for [MySQL SSL mode](#): [Disabled](#), [Preferred](#), [Required](#), [Verify_CA](#), or [Verify_Identity](#).

If you select [Disabled](#), there are no further SSL options to complete.

8. If you do not select [Disabled](#) for [MySQL SSL mode](#), complete the following options as applicable.

- [MySQL SSL key file](#)
- [MySQL SSL certificate file](#)
- [MySQL SSL CA file](#)

- [MySQL SSL CA path](#)
- [MySQL SSL Cipher](#)
- [MySQL SSL RSA public key file](#)

i Note

Find descriptions for the SSL options in [MySQL SSL encryption parameters for DSN-less connections \[page 158\]](#).

9. Restart the services as applicable.

Create the MySQL datastore following the steps in [Defining a database datastore \[page 96\]](#). See option descriptions for MySQL datastores at [MySQL datastore options \[page 146\]](#) and [MySQL SSL encryption parameters for DSN-less connections \[page 158\]](#).

Task overview: [SSL/TLS encryption for MySQL \[page 154\]](#)

Related Information

[Windows: Configuring DSN and SSL/TLS encryption for MySQL datastores \[page 155\]](#)

[MySQL SSL encryption parameters for DSN-less connections \[page 158\]](#)

[Using the Connection Manager](#)

9.1.10.5.3 MySQL SSL encryption parameters for DSN-less connections

Configure SSL/TLS protocol for MySQL DSN-less connections in the datastore editor in Designer.

When you create a MySQL datastore with DSN connections, you configure SSL/TLS protocol when you configure the DSN connection:

- Windows: Use the ODBC Data Source Administrator following the steps in [Windows: Configuring DSN and SSL/TLS encryption for MySQL datastores \[page 155\]](#).
- Linux and Solaris: Use the SAP Data Services Connection Manager following the steps in [Linux/Solaris: Configuring DSN and SSL/TLS encryption for MySQL datastores \[page 157\]](#).

However, to configure SSL/TLS protocol for MySQL datastores with DSN-less connections, use the datastore editor for all platforms.

The DSN and SSL/TLS options that you complete are similar regardless of the utility you use to create the connection. The following table contains descriptions for the SSL/TLS encryption options.

SSL/TLS encryption option descriptions

Option	Description
SSL Mode	Specifies the type of SSL/TLS mode Data Services uses to connect to the MySQL server. • Preferred: Permits but does not require encrypted connections. If client cannot connect using encryption, it uses an unencrypted connection. Preferred is the default setting. • Required: Requires encrypted connections. Connection fails if client cannot connect using an encrypted connection. • Verify_CA: Requires an encrypted connection and also performs verification against the server CA certificate. Connection fails if client cannot connect using an encrypted connection and-or the verification against the server CA certificate fails. • Verify_Identity: Requires an encrypted connection and also performs verification against the server host name in the CA certificate. Connection fails if client cannot connect using an encrypted connection and-or the verification against the server host name in the CA certificate fails. i Note If you complete this option in the SAP Data Services Connection Manager, the option is MySQL SSL mode. The option in Connection Manager has an additional value of Disabled, which specifies not to use SSL protocol.
SSL key	Specifies the path and file name of the server private key file. For example, <code>... \client_key.pem</code>. i Note In Connection Manager, the option is MySQL SSL key file.
SSL Certificate	Specifies the path and file name of the public key certificate file. For example, <code>... \client_cert.pem</code>. i Note In Connection Manager, the option is MySQL SSL certificate file.

Option	Description
<i>SSL CA File</i> In the ODBC Data Source Administrator, the option is SSL Certificate Authority.	Specifies the path and file name for the Certificate Authority certificate file. For example, <code>... \ca.pem</code>. i Note In Connection Manager, the option is <i>MySQL SSL CA file</i>.
<i>SSL CA Path</i>	Specifies the path for the directory of Certificate Authority certificate files. i Note In Connection Manager, the option is <i>MySQL SSL CA path</i>.
<i>SSL Cipher</i>	Specifies the list of permitted ciphers for the connection encryption. You designate the ciphers in the MySQL server configuration. If not applicable, leave blank. i Note In Connection Manager, the option is <i>MySQL SSL Cipher</i>.
<i>RSA Public Key</i>	Specifies the path and file name for the RSA public key file. (<code>... \public_key.pem</code>). Applicable only when you include RSA for <i>SSL Cipher</i>. i Note In Connection Manager, the option is <i>MySQL SSL RSA public key file</i>.

Parent topic: [SSL/TLS encryption for MySQL \[page 154\]](#)

Related Information

[Windows: Configuring DSN and SSL/TLS encryption for MySQL datastores \[page 155\]](#)

[Linux/Solaris: Configuring DSN and SSL/TLS encryption for MySQL datastores \[page 157\]](#)

9.1.11 ODBC

Create an ODBC database datastore to use in Data Services.

The following tables contain option descriptions specific to configuring an ODBC datastore.

Main window

Option	Description
Data source name	Select the Data Source Name (DSN) from the dropdown list.
ODBC Admin button	Click to launch the Windows ODBC Data Source Administrator. Select a supported driver for the database and configure a data source. After closing the ODBC Data Source Administrator, select the newly created data source name from the dropdown list. For Unix, use the DSCConnection Manager to create a DSN.

General

Option	Description
Database server working directory	Enter or browse for the working directory on the database server that stores files such as logs. Must be defined to use FTP.

FTP

i Note

If this datastore is not being used specifically for Netezza bulk loading, the software ignores any FTP option entries.

Option	Description
FTP host name	For a Netezza server, type the name of the Netezza server computer (host). Must be defined to use FTP. Use the computer name, fully qualified domain name, or IP address.
FTP login user name	Must be defined to use FTP.
FTP login password	Must be defined to use FTP.
FTP host working directory	The location on the database server from where the software retrieves diagnostic files generated by the database's bulk loader. It must be available from the FTP server. It is usually the same as the database's working directory. If unsure, contact your system administrator.

i Note

Configure the FTP server to accept an absolute path.

The options in the following table have one or more of the following values:

- Automatic
- ODBC syntax
- SQL-92 syntax
- Yes
- No

The default setting is [Automatic](#).

ODBC Capability Support

Option	Description
<i>Array fetch</i>	Select <i>No</i> if you encounter errors when reading from an ODBC datastore. For example, if you receive an error message that involves the ODBC call <code>SQLFetchScroll</code>, assume that your ODBC driver does not support array fetch. Selecting <i>No</i> turns off the array fetch capability. With the <i>Array fetch</i> option disabled, the software fetches one row at a time from the ODBC data source. The software does not display the <i>Array fetch size</i> performance option for any source table editors and SQL transform editors that use the ODBC datastore.
<i>Parameterized SQL</i>	The software generates SQL statements with parameters instead of literal values, which can significantly improve performance.
<i>Outer join</i>	Determines whether the ODBC driver supports outer join syntax.
<i>Auto commit</i>	Determines whether the ODBC driver supports auto commit.
<i>Server sorts in binary</i>	Determines whether the server performs binary sort.

ODBC Math Function Support

Option	Description
<i>Absolute</i>	Returns the absolute value of an input number.
<i>Ceiling</i>	Returns the smallest integer value greater than or equal to an input number.
<i>Floor</i>	Returns the largest integer value less than or equal to an input number.
<i>Round</i>	Rounds a given number to the specified precision.
<i>Truncate</i>	Truncates a given number to the specified precision.
<i>Sqrt</i>	Returns the square root of the input number.
<i>Log</i>	Returns the base-10 logarithm of the given numeric expression.
<i>Ln</i>	Returns the natural logarithm of the given numeric expression.
<i>Power</i>	Returns the value of the given expression to the specified power.
<i>Mod</i>	Returns the remainder when one number is divided by another.

ODBC String Function Support

Option	Description
<i>Lower case</i>	Changes the characters in a string to lowercase.
<i>Upper case</i>	Changes the characters in a string to uppercase.
<i>Rtrim blanks</i>	Removes blank characters from the end of a string.
<i>Ltrim blanks</i>	Removes blank characters from the start of a string.
<i>Length</i>	Returns the number of characters in a given string.
<i>Substring</i>	Returns a specific portion of a string starting at a given point in the string.
<i>Soundex</i>	Returns the soundex encoding of the input string.

ODBC Date Function Support

Option	Description
<i>System date</i>	Select the method to use to return the current date as listed by the Job Server operating system.
<i>System time</i>	Select the method to use to return the current time as listed by the operating system.
<i>Week</i>	Select the method to use to determine the week in the year in which the given date falls.
<i>Month</i>	Select the method to use to determine the month in which the given date falls.
<i>Quarter</i>	Select the method to use to determine the quarter in which the given date falls.
<i>Year</i>	Select the method to use to determine the year in which the given date falls.
<i>Day of month</i>	Select the method to use to determine the day in the month on which the given date falls.
<i>Day of year</i>	Select the method to use to determine the day in the year on which the given date falls.

ODBC Aggregate Function Support

Option	Description
<i>Average</i>	Calculates the average of a given set of values.
<i>Count</i>	Counts the number of values in a table column.
<i>Count Distinct</i>	Counts the number of distinct non-NULL values in a table column.
<i>Max</i>	Returns the maximum value from a list.
<i>Min</i>	Returns the minimum value from a list.
<i>Sum</i>	Calculates the sum of a given set of values.

Miscellaneous

Option	Description
<i>Date format</i>	Enter a date format supported by the data source (a date format that the data source ODBC driver and database supports). The default date format is <code>yyyymm-dd</code> .
<i>Time format</i>	Enter a time format supported by the data source (a time format that the data source ODBC driver and database supports). The default time format is <code>hh:mi:ss</code> .
<i>Date-time format</i>	Enter a date/time format supported by the data source (a date/time format supported by the data source ODBC driver and database). The default date/time format is <code>yyyymm-dd hh:mi:ss</code> .
<i>Decimal separator</i>	Enter the character that the data source uses to separate the decimal portion of a number. The default is a period.
<i>Data type conversion support</i>	When there's a data type mismatch in an expression, the software automatically generates an explicit convert function call.
<i>NVL support</i>	If the input value is NULL, replace with the specified value.
<i>Iffthenelse support</i>	No: Does not allow conditional logic in mapping and selection operations. This is the default. Yes: Allows conditional logic in mapping and selection operations.

Option	Description
<i>NVARCHAR type name</i>	Select an option for loading multibyte data to template tables based on the database type: • <i><Unknown></i>: Select when you do not know the type name. • <i>NVARCHAR</i>: Select for all supported databases except Oracle. • <i>NVARCHAR2</i>: Select for Oracle databases only. In the template table target editor, setting the option <i>Use NVARCHAR for VARCHAR columns in supported databases</i> to Yes enables this data type conversion.

[ODBC datastore option value descriptions \[page 164\]](#)

The ODBC database datastore has specific ODBC options for which you select values from a dropdown list.

Parent topic: [Database datastores \[page 93\]](#)

Related Information

[Defining a database datastore \[page 96\]](#)
[Datastore metadata \[page 102\]](#)
[Imported metadata from database datastores \[page 107\]](#)
[Attunity Connector \(mainframe datastore\) \[page 115\]](#)
[Data Federator \[page 121\]](#)
[IBM DB2 datastores \[page 124\]](#)
[Informix \[page 127\]](#)
[Microsoft SQL Server datastore \[page 129\]](#)
[Memory datastores \[page 136\]](#)
[MySQL datastore \[page 145\]](#)
[Oracle datastores \[page 166\]](#)
[Netezza \[page 170\]](#)
[SAP SQL Anywhere \[page 172\]](#)
[SAP ASE \[page 173\]](#)
[Sybase IQ \[page 174\]](#)
[Teradata \[page 179\]](#)
[ODBC datastore option value descriptions \[page 164\]](#)
[Common datastore options \[page 99\]](#)

9.1.11.1 ODBC datastore option value descriptions

The ODBC database datastore has specific ODBC options for which you select values from a dropdown list.

To define an ODBC datastore connection, define a data source name (DSN), a user name, a password if applicable, and optionally a set of advanced options.

Selecting an ODBC data source name

Select a data source name in one of the following ways in the [Data source name](#) option of the ODBC datastore editor:

- Select an existing DSN from the [Data Source Name](#) dropdown list.
- Type the name of an existing DSN.
- Click [ODBC Admin](#) to launch the Windows ODBC Data Source Administrator where you create or configure data sources. After closing the ODBC Data Source Administrator, select the newly created DSN from the [Data Source Name](#) dropdown list.

Defining ODBC datastore options

Click [Advanced](#) to open the advanced options for defining an ODBC datastore. For each applicable option, select a value from the dropdown list or type a custom value when allowed.

The following table contains descriptions for the ODBC datastore options that appear in the dropdown lists.

Value	Description
Automatic	For most options, this is the default value. Select Automatic if you do not know if the ODBC driver supports the option. With Automatic selected, SAP Data Services queries the driver to determine its capabilities: • If the driver supports that option, Data Services pushes down the operation to the ODBC database. • If the ODBC driver does not support that option, Data Services executes the operation internally. To circumvent possible inconsistencies with the ODBC driver, specify an option other than Automatic. If you select anything other than Automatic, the software does not query the driver for that particular capability.
ODBC syntax	The software assumes that the ODBC driver supports the option function or capability and uses ODBC syntax. For example, for the ABSOLUTE function, the ODBC syntax is: <pre>{fn abs (TAB1.COL1)}</pre>
SQL-92	The software assumes that the ODBC driver supports the option function or capability and uses SQL-92 syntax. For example, when the software generates an explicit CONVERT function, the SQL-92 syntax is: <pre>CAST (TAB1.VC_COL AS SQL_INTEGER)</pre>
No	The software assumes that the ODBC driver does not support the option function or capability and executes it internally.

Value	Description
Custom	Many functions allow you to type in the specific function call to use for that option. The software assumes that the ODBC driver supports the function capability. Note You cannot specify the signature of the function; it is the same as in the ODBC signature. For example, for the string function <i>Upper case</i>, instead of using { fn ucase (. . .) }, type in the <i>Upper case</i> option field upper. The software generates the following: <pre>upper (TAB1.VC_COL)</pre>

Parent topic: [ODBC \[page 161\]](#)

Related Information

[Configure drivers with data source name \(DSN\) connections](#)

9.1.12 Oracle datastores

To use Oracle data as sources and targets in data flows, create an Oracle database datastore.

SAP Data Services supports both Transport Network Substrate (TNS) and server-name (TNS-less) connections in Oracle datastores. Additionally, use SSL/TLS encryption with datastores that have TNS-less connections.

Before you connect to your Oracle databases with a datastore, download, install, and configure the applicable Oracle driver. For specific information about the applicable driver version for your Oracle installation, see the [Product Availability Matrix \(PAM\)](#) on the SAP Support Portal.

Configure the driver using the utility for your platform:

- Windows: ODBC Drivers Selector
- Unix: SAP Data Services Configuration Manager

Additionally, if you plan to use a TNS connection, create the TNS following the directions in your Oracle documentation.

The Oracle Advanced Security Package is included in your Oracle installation. The Advanced Security Package has the required tools for configuring SSL/TLS encryption. For complete information about using SSL/TLS for Oracle, see the *Oracle Advanced Security Users Guide*. See your Oracle documentation for instructions to create an SSL, which includes the certificate and public and private key pair. Oracle stores this data in your wallet.

[Oracle database datastore options \[page 167\]](#)

Create an Oracle database datastore and set options for the connection type and SSL/TLS protocol when applicable.

Parent topic: [Database datastores \[page 93\]](#)

Related Information

[Defining a database datastore \[page 96\]](#)
[Datastore metadata \[page 102\]](#)
[Imported metadata from database datastores \[page 107\]](#)
[Attunity Connector \(mainframe datastore\) \[page 115\]](#)
[Data Federator \[page 121\]](#)
[IBM DB2 datastores \[page 124\]](#)
[Informix \[page 127\]](#)
[Microsoft SQL Server datastore \[page 129\]](#)
[Memory datastores \[page 136\]](#)
[MySQL datastore \[page 145\]](#)
[ODBC \[page 161\]](#)
[Netezza \[page 170\]](#)
[SAP SQL Anywhere \[page 172\]](#)
[SAP ASE \[page 173\]](#)
[Sybase IQ \[page 174\]](#)
[Teradata \[page 179\]](#)
[Using the Connection Manager](#)
[Using the ODBC Drivers Selector for Windows](#)

9.1.12.1 Oracle database datastore options

Create an Oracle database datastore and set options for the connection type and SSL/TLS protocol when applicable.

The following table contains basic options and values for an Oracle datastore.

Main window options

Option	Description
Datastore Type	Select Database.
Database Type	Select Oracle.

Option	Description
Change data capture (CDC) options dropdown list	Available options are based on the Oracle version you select. For all versions except Oracle 12c: • No CDC: Disables change data capture. • Native CDC: Uses Microsoft SQL Server change data capture. • Replication Server CDC: Uses the Replication Server change data capture. For Oracle 12c: • No CDC • Replication Server CDC
Use TNS name	Specifies whether to use a Transparent Network Substrate (TNS) connection. Select to use a TNS connection. Create a TNS connection before you create the datastore. See your Oracle documentation for instructions to create a TNS. Leave unselected to use a server-name (TNS-less) connection. When you leave unselected, the SSL options under Advanced become enabled.
The following options appear when you select to create a TNS connection:	
TNS Name	Enter an existing Oracle Transparent Network Substrate (TNS) name through which Data Services accesses the Oracle database.
User Name	Specifies the user name authorized to access the database specified in the TNS connection.
Password	Specifies the password associated with the user name.
The following options appear when you select to create a TNS-less connection:	
Hostname	Specifies the computer name, fully qualified domain name, or IP address of the machine where the Oracle Server instance is located.
SID or Service Name	Specifies the System ID for the Oracle database.
Port	Specifies the port number to access the Oracle database.
User Name	Specifies the name of the user authorized to access the database.
Password	Specifies the password associated with the user name.

The following table describes advanced options specific to Oracle datastores.

Option	Description
<i>Use SSL encryption</i>	Specifies to use SSL/TLS encryption for this datastore. Applicable only for TNS-less connections. → Remember Ensure that you complete all prerequisites for using SSL/TLS encryption for Oracle before you create the datastore.
<i>Encryption Parameters</i>	Opens the <i>Encryption Parameters</i> dialog box. Double-click in the blank cell to open the <i>Encryption Parameters</i> dialog box.
The following options appear in the <i>Encryption Parameters</i> dialog box:	
<i>Validate Certificate</i>	Specifies to validate the server certificate using the information stored in the Oracle wallet.
<i>Distinguished name</i>	Specifies the Oracle database connection DN for the database. Applicable when you configure the Oracle Net Service name to include server DN matching and to use TCP/IP with SSL/TLS on the client. ! Restriction If you entered <code>SSL_SERVER_DN_MATCH=ON</code> in the <code>sqlnet.ora</code> file, The DN that you enter in <i>Distinguished name</i> must match the DN that you entered when you created the SSL certificate. For complete information about using the optional Server DN matching, see the <i>Oracle Database Security Guide</i>.
<i>Certificate directory</i>	Specifies the location of the client-side wallet. Complete only when you select the <i>Validate Certificate</i> option.
<i>Certificate password</i>	Specifies the password to log in to the client side wallet. Complete only when you select the <i>Validate Certificate</i> option.
The following options are under <i>Oracle Miscellaneous</i>:	
<i>Default precision for number</i>	Specifies the total number of digits in a numeric value. Enter a value from 1 to 96.

Option	Description
<i>Default scale for number</i>	Specifies the number of digits to the right of the decimal point in a numeric value. For example, 0 <= scale <= precision.

Related Information

[Oracle TCPS protocol for repositories](#)

9.1.13 Netezza

Create a Netezza database datastore for Data Services.

Main window

Option	Description
<i>Data source name</i>	Enter the data source name defined in Netezza for connecting to your database. This option is required when you select Use data source name (DSN). Create a DSN by clicking ODBC Admin or by using the DSCONNECTION Manager (Unix). For information about creating a DSN with a supported driver, see the <i>Designer Guide</i>. If you use the Auto correct load feature for Netezza targets, be sure that your data source allows your user name to create or replace stored procedures.
<i>Enable automatic data transfer</i>	The Enable automatic data transfer checkbox is selected by default when you create a new datastore and you chose Database for Datastore type. Keep Enable automatic data transfer selected to enable transfer tables in this datastore that the Data_Transfer transform can use to push down subsequent database operations.

General

Option	Description
<i>Database Server working directory</i>	Enter or browse to the location of the working directory for the load utility on the computer that runs the Netezza server. Complete this option whenever the Netezza server and the Job Server run on separate machines.

FTP

Option	Description
<i>FTP host name</i>	Enter the FTP host name using the computer name, fully qualified domain name, or IP address. If you leave this field blank or you enter the name of the computer (host) where the Job Server resides, the software assumes that MySQL and the software share the same computer and that FTP is unnecessary. When FTP is unnecessary, all other FTP-related fields can remain blank.
<i>FTP login user name</i>	Must be defined to use FTP.
<i>FTP login password</i>	Must be defined to use FTP.
<i>FTP host working directory</i>	Enter the absolute file path to the location on the database server where Data Services transfers the data file between the Job Server and the MySQL server. For Windows servers only, you can configure a path to a virtual directory.

ODBC Miscellaneous

Option	Description
<i>Date format</i>	Enter a date format supported by the data source (a date format that the data source ODBC driver and database supports). The default date format is <code>yyyy-mm-dd</code> .
<i>Time format</i>	Enter a time format supported by the data source (a time format that the data source ODBC driver and database supports). The default time format is <code>hh:mi:ss</code> .
<i>Date-time format</i>	Enter a date/time format supported by the data source (a date/time format supported by the data source ODBC driver and database). The default date/time format is <code>yyyy-mm-dd hh:mi:ss</code> .
<i>Decimal separator</i>	Enter the character that the data source uses to separate the decimal portion of a number. The default is a period.
<i>Data type conversion support</i>	When there's a data type mismatch in an expression, the software automatically generates an explicit convert function call. Options are: • <i>Automatic</i> (default) • <i>ODBC syntax</i> • <i>SQL-92 syntax</i> • <i>No</i> See ODBC datastore option value descriptions [page 164] for value descriptions.
<i>NVL support</i>	If the input value is NULL, replace with the specified value. Options are: • <i>Automatic</i> (default) • <i>ODBC syntax</i> • <i>No</i> (disables this feature) See ODBC datastore option value descriptions [page 164] for value descriptions.
<i>Ifthenelse support</i>	<i>No</i>: Does not allow conditional logic in mapping and selection operations. This is the default. <i>Yes</i>: Allows conditional logic in mapping and selection operations.

Parent topic: [Database datastores \[page 93\]](#)

Related Information

[Defining a database datastore \[page 96\]](#)
[Datastore metadata \[page 102\]](#)
[Imported metadata from database datastores \[page 107\]](#)
[Attunity Connector \(mainframe datastore\) \[page 115\]](#)
[Data Federator \[page 121\]](#)
[IBM DB2 datastores \[page 124\]](#)
[Informix \[page 127\]](#)
[Microsoft SQL Server datastore \[page 129\]](#)
[Memory datastores \[page 136\]](#)
[MySQL datastore \[page 145\]](#)
[ODBC \[page 161\]](#)
[Oracle datastores \[page 166\]](#)
[SAP SQL Anywhere \[page 172\]](#)
[SAP ASE \[page 173\]](#)
[Sybase IQ \[page 174\]](#)
[Teradata \[page 179\]](#)
[Common datastore options \[page 99\]](#)

9.1.14 SAP SQL Anywhere

Create an SAP SQL Anywhere database datastore to use in Data Services.

The following table contains descriptions specific to SAP SQL Anywhere database configuration.

Main window

Option	Description
<i>Enable Automatic Data Transfer</i>	Enables transfer tables in this datastore, which the Data_Transfer transform can use to push down subsequent database operations. This option is enabled by default.

Parent topic: [Database datastores \[page 93\]](#)

Related Information

[Defining a database datastore \[page 96\]](#)
[Datastore metadata \[page 102\]](#)
[Imported metadata from database datastores \[page 107\]](#)

[Attunity Connector \(mainframe datastore\) \[page 115\]](#)
[Data Federator \[page 121\]](#)
[IBM DB2 datastores \[page 124\]](#)
[Informix \[page 127\]](#)
[Microsoft SQL Server datastore \[page 129\]](#)
[Memory datastores \[page 136\]](#)
[MySQL datastore \[page 145\]](#)
[ODBC \[page 161\]](#)
[Oracle datastores \[page 166\]](#)
[Netezza \[page 170\]](#)
[SAP ASE \[page 173\]](#)
[Sybase IQ \[page 174\]](#)
[Teradata \[page 179\]](#)
[Common datastore options \[page 99\]](#)

9.1.15 SAP ASE

To use SAP ASE metadata in SAP Data Services, create an SAP ASE database datastore.

The following table contains descriptions specific for SAP ASE datastore configuration.

Main window

Option	Description
<i>CDC options</i>	Select an option from the dropdown list. Options include the following: <i>No CDC</i>: Disables change data capture. <i>Replication Server CDC</i>: Uses the Replication Server change data capture.
<i>Database server name</i>	Enter the name of the computer where the SAP ASE instance is located. i Note For Unix Job Servers, when you log in to an SAP Sybase repository in the Designer, the case in which you type the database server name must match the associated case in the SYBASE_Home\interfaces file. If the case does not match, you might receive an error because the Job Server cannot communicate with the repository.

Parent topic: [Database datastores \[page 93\]](#)

Related Information

[Defining a database datastore \[page 96\]](#)
[Datastore metadata \[page 102\]](#)

[Imported metadata from database datastores \[page 107\]](#)
[Attunity Connector \(mainframe datastore\) \[page 115\]](#)
[Data Federator \[page 121\]](#)
[IBM DB2 datastores \[page 124\]](#)
[Informix \[page 127\]](#)
[Microsoft SQL Server datastore \[page 129\]](#)
[Memory datastores \[page 136\]](#)
[MySQL datastore \[page 145\]](#)
[ODBC \[page 161\]](#)
[Oracle datastores \[page 166\]](#)
[Netezza \[page 170\]](#)
[SAP SQL Anywhere \[page 172\]](#)
[Sybase IQ \[page 174\]](#)
[Teradata \[page 179\]](#)
[Change data capture \(CDC\) \[page 101\]](#)
[Common datastore options \[page 99\]](#)

9.1.16 Sybase IQ

To view, browse, and import data from SAP IQ, create a database datastore.

i Note

SAP Sybase IQ is now known as SAP IQ.

The following tables contain descriptions specific for creating an SAP IQ database datastore. For common datastore descriptions, see [Common datastore options \[page 99\]](#).

Sybase IQ datastore option descriptions

Option	Description
<i>Database Type</i>	Select Sybase IQ
Advanced options, Connection group	
<i>Additional connection parameters</i>	Optional. To configure SAP Data Services to display multi-byte data to display correctly from SAP IQ UTF-8 databases, enter charset=utf8 . Applies to datastores connecting with a server-based (DSN-less) connection.
Bulk Loader group	

Option	Description
<i>JS and DB on same machine</i>	Specifies whether Data Services transfers the data file generated on the Job Server (JS) to the database server (DB) using FTP protocol. • Yes: Transfers the data file generated on the Job Server to the database server using FTP protocol. • No: Does not transfer the data file generated on the Job Server to the database server using FTP protocol. No is the default setting. ! Restriction For more information about how to use this option with the <i>Use named pipe</i> option and the FTP options based on the SAP IQ version, see Option considerations for SAP Sybase IQ (SAP IQ) bulk loading [page 178].
<i>Server working directory</i>	Specifies the path to the working directory for the load utility, which is the computer that runs the SAP IQ server. Special considerations: • If you enter an absolute file path for the <i>FTP host working directory</i> option in the FTP option group, then this option (<i>Server working directory</i>) is optional. • If you leave this option (<i>Server working directory</i>) blank, then the <i>FTP host working directory</i> option is required. • If you enter a virtual file path in the <i>FTP host working directory</i> option, you must enter an absolute path in this option (<i>Server working directory</i>).

Option	Description
<i>Use named pipe</i>	Specifies whether Data Services uses a named pipe for data transfer. • Yes: Uses a named pipe to transfer the data file. Selecting Yes may improve performance. • No: Writes data file to a disk to transfer the data file. No is the default setting. i Note If SAP IQ database recovery requires a data file, select No for <i>Use named pipe</i>. ! Restriction For more information about how to use this option with the <i>JS and DB on same machine</i> option and the FTP options based on the SAP IQ version, see Option considerations for SAP Sybase IQ (SAP IQ) bulk loading [page 178].
FTP option group	
<i>FTP host name</i>	Specifies the host name of the FTP server (host) computer. Specify the host name when you want Data Services to transfer the data file using FTP to the database for loading. You can leave this option blank. However, if Data Services needs the FTP host name information for bulk loading, and you do not enter a value for <i>FTP host name</i>, Data Services issues a validation error.
<i>FTP login user name</i>	Specifies the user name to log into FTP server. Required for FTP.
<i>FTP login password</i>	Specifies the password related to the login user name to access the FTP server. Required for FTP.

Option	Description
FTP host working directory	Specifies the location on the database server where Data Services transfers the data file between the Job Server and the SAP IQ server. For Windows servers, configure a path to a virtual directory. Special considerations: • If you enter an absolute file path for this option (FTP host working directory), then the option Server working directory in the Bulk loader group is optional. • If you leave the Server working directory option blank, then this option (FTP host working directory) is required. • If you enter a virtual file path in this option (FTP host working directory), you must enter an absolute path in the Server working directory option.
Sybase Miscellaneous group	
Use linked remote servers	Specifies whether to use a linked remote server when you use the SQL statement <code>INSERT_LOCATION</code> for a data flow that: • Uses SAP IQ as the loader. • Uses either SAP ASE or SAP IQ as the reader. Data Services pushes the SQL statement down for the SAP IQ server location. • Yes: Uses remote servers that are already linked. • No: Does not use remote servers that are already linked. To define a remote server, use the <code>CREATE SERVER</code> SQL statement in SAP IQ. To set up the remote logon for users, use the <code>CREATE EXTERNLOGIN</code> SQL statement. For detailed information about the SQL statements, see the relevant SAP IQ product documentation.

Parent topic: [Database datastores \[page 93\]](#)

Related Information

[Defining a database datastore \[page 96\]](#)

[Datastore metadata \[page 102\]](#)

[Imported metadata from database datastores \[page 107\]](#)

[Attunity Connector \(mainframe datastore\) \[page 115\]](#)

[Data Federator \[page 121\]](#)
[IBM DB2 datastores \[page 124\]](#)
[Informix \[page 127\]](#)
[Microsoft SQL Server datastore \[page 129\]](#)
[Memory datastores \[page 136\]](#)
[MySQL datastore \[page 145\]](#)
[ODBC \[page 161\]](#)
[Oracle datastores \[page 166\]](#)
[Netezza \[page 170\]](#)
[SAP SQL Anywhere \[page 172\]](#)
[SAP ASE \[page 173\]](#)
[Teradata \[page 179\]](#)
[Common datastore options \[page 99\]](#)

9.1.16.1 Option considerations for SAP Sybase IQ (SAP IQ) bulk loading

When you configure the SAP IQ datastore, consider the SAP IQ version number before setting options that affect bulk loading.

Datastore options that you set based on your version of SAP IQ for bulk loading are in the Bulk Loader group in the Advanced options of the datastore editor. The following table describes the options.

Bulk Loader group option descriptions

Option	Description
<i>JS and DB on same machine</i>	Specifies whether Data Services transfers the data file generated on the Job Server (JS) to the database server (DB) using FTP protocol.
<i>Use named pipe</i>	Specifies whether Data Services uses a named pipe for data transfer. If set to <i>No</i> , Data Services writes data file to a disk to transfer the data file.

The following table contains settings to make based on your version of SAP IQ. It includes the method of bulk loading to use, the values for the SAP IQ datastore options *JS and DB on same machine* and *Use named pipe*, and the applicable SAP IQ version.

Supported bulk loader options and methods

SAP IQ version	Method	Option JS and DB on same machine	Option Use named pipe
Earlier than 15.x	Named pipe	Yes	Yes
	File	Yes	No
	FTP	No	Yes
	FTP	No	No

SAP IQ version	Method	Option JS and DB on same machine	Option Use named pipe
15.x and later	Named pipe	Yes	Yes
	File	Yes	No
	Named pipe	No	Yes
	File	No	No

❖ Example

If you use a version of SAP IQ **earlier than 15.x**, and you use the **named pipe** method, select **Yes** for the bulk loading options *JS and DB on same machine* and *Use named pipe*.

9.1.17 Teradata

Create a Teradata database datastore to use in Data Services.

The following tables contain descriptions of options specific to the Teradata datastore.

General

Teradata option	Description
<i>Bulk reader directory</i>	Enter or browse to the location of the directory on the Job Server where you store Teradata Parallel Transporter files (control and error files). If you leave this option empty, the software uses the default location <code><LINK_DIR>\log\BulkReader</code> .

Locale

Teradata option	Description
<i>Teradata</i>	
<i>Log directory</i>	Enter or browse to the directory in which to write log files.
<i>TdpId</i>	Enter the Teradata Director Program Identifier (TdpId) that identifies the name of the Teradata database to load. Required when you use bulk loading.

Teradata option	Description
<i>Additional session parameters</i>	Additional session parameters specified as a valid SQL statement. Delimit multiple SQL statements with a semicolon.

❖ Example

For example, to use the Table_Comparison transform with Teradata 13 and later tables as the comparison table and target table, make the following settings:

- On the Teradata server, set the *General* parameter *DBSControl* to TRUE to allow uncommitted data to be read.
- In the Data Services Teradata datastore, add the following statement in *Additional session parameters*:

```
SET SESSION CHARACTERISTICS AS TRANSACTION
ISOLATION LEVEL READ UNCOMMITTED;
```

Parent topic: [Database datastores \[page 93\]](#)

Related Information

[Defining a database datastore \[page 96\]](#)
[Datastore metadata \[page 102\]](#)
[Imported metadata from database datastores \[page 107\]](#)
[Attunity Connector \(mainframe datastore\) \[page 115\]](#)
[Data Federator \[page 121\]](#)
[IBM DB2 datastores \[page 124\]](#)
[Informix \[page 127\]](#)
[Microsoft SQL Server datastore \[page 129\]](#)
[Memory datastores \[page 136\]](#)
[MySQL datastore \[page 145\]](#)
[ODBC \[page 161\]](#)
[Oracle datastores \[page 166\]](#)
[Netezza \[page 170\]](#)
[SAP SQL Anywhere \[page 172\]](#)
[SAP ASE \[page 173\]](#)
[Sybase IQ \[page 174\]](#)
[Common datastore options \[page 99\]](#)

9.2 Persistent cache datastores

A persistent cache datastore is a container for cache tables.

Persistent cache tables allow you to cache large amounts of data. Persistent cache tables cache data from relational database tables and files.

i Note

You cannot cache data from hierarchical data files and files that contain nested schemas such as XML messages and SAP IDocs. You cannot perform incremental inserts, deletes, or updates on a persistent cache table.

A persistent cache datastore contains cache table schemas saved in your repository. In contrast, a regular database datastore provides a connection to a database, application, or adapter.

The following table describes the benefits of using persistent cache datastores for data flows that process large volumes of data.

Advantage	Example
Store large amounts of data in persistent cache. The software quickly loads data into memory to provide immediate access during a job.	Use persistent cache tables to access a lookup table or comparison table locally instead of from a remote database.
Use cache tables in multiple data flows.	Use a persistent cache table for a large lookup table in a lookup_ext function that rarely changes. Create a cache once. Then subsequent jobs use this cache instead of creating a new cache each time.

i Note

For real-time jobs, consider using memory datastores that contain memory tables.

[Defining a persistent cache datastore \[page 182\]](#)

Create and define a persistent cache datastore in the same manner as creating a database datastore.

[Persistent Cache datastore options \[page 182\]](#)

Option descriptions for setting up a persistent cache datastore.

[Create persistent cache tables \[page 183\]](#)

There are two methods to create a persistent cache table.

[Use persistent cache tables \[page 186\]](#)

Use a persistent cache table as a source, lookup table, or comparison table.

Parent topic: [Datastores \[page 92\]](#)

Related Information

[Database datastores \[page 93\]](#)
[Linked datastores \[page 186\]](#)
[Adapter datastores \[page 190\]](#)
[Application datastores \[page 191\]](#)
[Web service datastores \[page 194\]](#)
[Change a datastore definition \[page 195\]](#)
[Create and manage multiple datastore configurations \[page 197\]](#)

9.2.1 Defining a persistent cache datastore

Create and define a persistent cache datastore in the same manner as creating a database datastore.

1. Select **Project** > **New** > **Datastore**.

The datastore editor opens.

2. Type a name for the datastore in the **Name** text box.

If you have adopted Data Services naming conventions, ensure that you prefix the datastore name with **Persist_DS**. The software appends datastore names to table names when you add a persistent cache table to a data flow in the workspace. The software uses regular table icons to represent persistent cache tables. Therefore, label a persistent cache datastore to distinguish the memory tables from regular database tables in the workspace.

3. Keep the default setting of **Database** for **Datastore type**.
4. Select **Persistent cache** from the **Database Type** dropdown list.
5. Enter the directory or browse to the directory in the **Cache directory** text box.

The directory is where the software stores the persistent cache for this datastore.

6. Click **OK**.

The software creates the persistent cache datastore. The new datastore appears alphabetically in the **Datastore** tab of the object library.

9.2.2 Persistent Cache datastore options

Option descriptions for setting up a persistent cache datastore.

The following table contains option descriptions specific to Persistent Cache datastores.

Basic options

Option	Description
Cache directory	Enter or browse to the location of your cache directory.

Option	Description
Force UTF-16 codepage	Affects how a dataflow runs when the datastore is used in a dataflow. - Selected: Caches data in a Unicode multi-byte format for persistent cache database types only. - Not selected: Runs the dataflow based on the engine runtime locale. You may benefit by selecting this option when you have several data flows that are dependent on each other and you have a mix of single-and multi-byte data. Setting this option ensures that the dependent data flows are compatible. However, if you have single-byte data and you select this option, you may see performance degradation in the data flow that uses this persistent cache.

Related Information

[Common datastore options \[page 99\]](#)

9.2.3 Create persistent cache tables

There are two methods to create a persistent cache table.

When you create a persistent cache table, you do not have to specify the table schema or import the table metadata. Instead, Data Services automatically creates the schema for each persistent cache table based on the preceding schema.

The first time you save the job that contains the persistent cache table, the software defines the table schema and saves the table. Then, the table appears with a table icon in the workspace and in the object library under the related persistent cache datastore.

You create a persistent cache table in one of the following ways:

- From a target template table in a data flow
- With a Data_Transfer transform during job execution

Related Information

Reference Guide: Data_Transfer

9.2.3.1 Creating a persistent cache table from a target template table

Create a persistent cache table from a template table that you add as a target in a data flow.

Open an existing batch job or begin to create a batch job in the workspace.

1. Click the template table icon  in the tool palette and click in the workspace.

The *Create Template* dialog box opens.

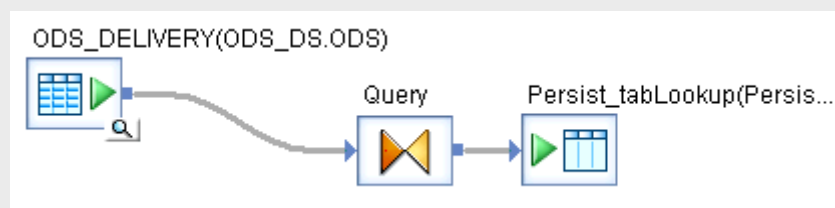
2. Enter a table name in the *Template name* text box.
3. Select the applicable persistent cache datastore from the *In datastore* dropdown list.
4. Select *Quote names* if applicable.
5. Click *OK*.

The persistent cache table appears in the workspace as a template table icon.

6. Connect the persistent cache table to the data flow as a target. (usually a Query transform).

❖ Example

The object that preceeds the persistent cache table is a *Query* transform in the following diagram:



Open the upstream object in the data flow and start creating the schema for the persistent cache schema following the steps in [Creating a persistent cache table schema in a data flow \[page 184\]](#).

Related Information

[Target persistent cache tables](#)

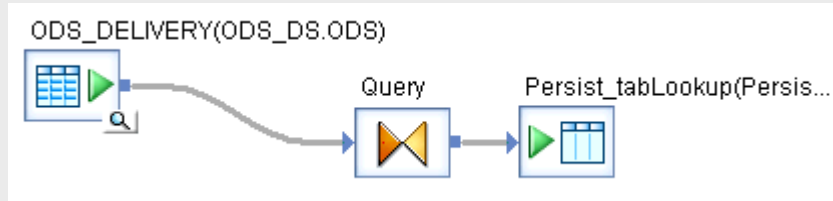
9.2.3.2 Creating a persistent cache table schema in a data flow

Create the persistent cache table schema in the work flow starting with the upstream object.

Follow the steps in [Creating a persistent cache table from a target template table \[page 184\]](#). Use the following diagram as an example:

❖ Example

The object that precedes the persistent cache table is a Query transform in the following diagram:



After you connect the persistent cache target table to the data flow, perform the following steps:

1. Open the object editor for the object that directly precedes the persistent cache table.

In the example, it is the Query transform editor.

2. Select the columns to include in the persistent cache table.
3. Drag the columns from the *Schema In* pane to the *Schema Out* pane.
4. Click the name of the persistent cache table to open the editor.
5. Set options as described in the following table.

Option	Description
<i>Column comparison</i>	Specifies how the software maps the input columns to persistent cache table columns. ◦ <i>Compare by position</i>: The software disregards the column names and maps source columns to target columns by position. ◦ <i>Compare by name</i>: The software maps source columns to target columns by name. <i>Compare by name</i> is the default setting.
<i>Include duplicate keys</i>	Specifies whether the software caches duplicate keys. Selected: The software caches duplicate keys. Selected is the default setting. Not selected: The software does not cache duplicate keys.

6. Open the *Keys* tab to specify the key column or columns to use as the key or keys in the persistent cache table.
7. Select ► *Project* ► *Save*. ►

The persistent cache template table icon changes to a target table icon. The software lists the table in the object library in the Table node under the applicable persistent cache datastore.

9.2.4 Use persistent cache tables

Use a persistent cache table as a source, lookup table, or comparison table.

After you create a persistent cache table as a target in one data flow, you can use the persistent cache table as a source in any data flow.

Related Information

[Reference Guide: Persistent cache source](#)

[Lookup tables and the lookup_ext function \[page 288\]](#)

9.3 Linked datastores

Linked datastores are datastores in a database link relationship.

The definition of a database link varies based on the database type:

- Oracle defines database links as one-way communication paths from one database server to another database server.
- DB2 defines database links like Oracle except an information server provides the one-way communication path. The information server allows a set of servers to get data from remote data sources.
- Microsoft SQL Server uses linked servers to provide the one-way communication path from one database server to another.

SAP Data Services refers to communication paths between databases as “database links.” Database links allow local users to access data on a remote database. The remote database can be on the local or a remote computer. The remote database can be of the same or different database type as the local database.

❖ Example

You have a local Oracle database server named ORDERS. ORDERS stores a database link that accesses information from a remote Oracle database named CUSTOMERS.

ORDERS: Oracle database on a local server

CUSTOMERS: Oracle database on a remote server

Users who are connected to the CUSTOMERS database cannot use the same database link that users connected to ORDERS use to connect to CUSTOMERS. The connection from ORDERS to CUSTOMERS is one way. Therefore, users who are connected to the CUSTOMERS database define a separate link to access the ORDERS database. They store the link in the data dictionary of the CUSTOMERS database.

Data Services uses linked datastores to enhance performance by pushing down operations to a target database using a target datastore. The advantages of pushing down operations to a database include:

- Reduces the amount of information to transfer between the databases and SAP Data Services
- Takes advantage of the various database management systems capabilities, such as various join algorithms.

[Relationship between database links and datastores \[page 187\]](#)

Database links and datastores contain the same types of information, including the same remote data source connection information.

[Linking a target datastore to a source datastore using a database link \[page 189\]](#)

A database link stores information about how to connect to a remote data source.

Parent topic: [Datastores \[page 92\]](#)

Related Information

[Database datastores \[page 93\]](#)

[Persistent cache datastores \[page 181\]](#)

[Adapter datastores \[page 190\]](#)

[Application datastores \[page 191\]](#)

[Web service datastores \[page 194\]](#)

[Change a datastore definition \[page 195\]](#)

[Create and manage multiple datastore configurations \[page 197\]](#)

9.3.1 Relationship between database links and datastores

Database links and datastores contain the same types of information, including the same remote data source connection information.

Database links and datastores store the following types of information:

- Host name
- Database name
- User name
- Password
- Database type

When you associate two datastores and use the option to import an external database link, each datastore connects to the databases defined in the database link. The following list includes additional requirements for creating linked datastores:

- The local server defined in the database link is a target server in Data Services.
- The remote server defined in the database link must be a source server in Data Services.
- An external database link, which exists first in a database, establishes the relationship between any target datastore and a source datastore.
- A Local datastore can be related to zero datastores or multiple datastores. If related to multiple datastores, there is a database link for each remote database.
- Two datastores can be related to each other with only one link.

The following diagram shows the possible relationships between database links (DBLink) and linked datastores (DS):

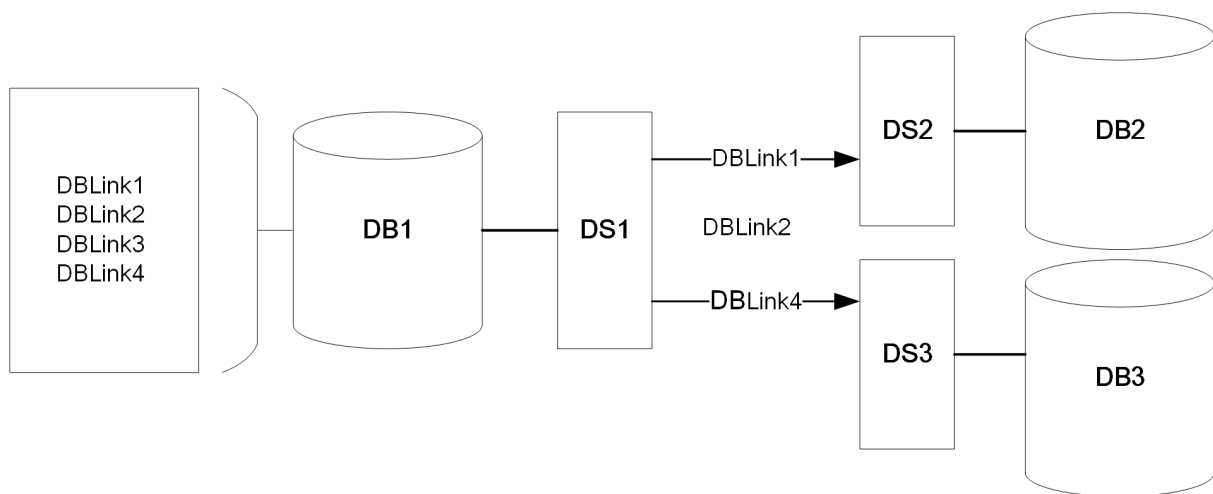


Diagram explanation

DB1 on the local server has four database links: DBLink1 to DBLink4. Data Services reads the database links in datastore DS1. The following table describes how DBLink1-4 relate.

DBLink	Relationship	Information
DBLink1	Maps DS1 to DS2	This relationship is called "linked data-store DBLink1". Data Services names the linked datastore with the same name as the external DBLink.
DBLink2	Relates DS1 to DS2	Does not map any datastore in Data Services but it relates datastore DS1 to DS2. These two datastores are also related by DBLink1. Although it is not usual, you can create multiple external database links that connect to the same remote source. However, Data Services allows only one database link between a target data-store and a source datastore pair.
DBLink3	None	Does not map to any DS in the software because there is no datastore defined for the remote data source to which the external database link refers.

❖ Example

If you link DBLink1 to target data-store DS1 with source datastore DS2, you cannot import DBLink2 to do the same.

DBLink	Relationship	Information
DBLink4	Maps DS1 to DS3	This relationship is called "linked data-store DBLink4". Data Services names the linked datastore with the same name as the external DBLink.

9.3.2 Linking a target datastore to a source datastore using a database link

A database link stores information about how to connect to a remote data source.

Perform the following steps to import a database link to a target datastore:

1. From the [Datastores](#) tab in the object library, right-click a target datastore and select [Edit](#).

If the database type supports database links, the list of configuration options includes the [Linked Datastores](#) option, in the [Advanced](#) section.

i Note

The datastore editor allows you to edit database links on target datastores for the default configuration only. So, if your target datastore contains multiple configurations (for example: Config1, Config2, and Config3), change the default to the applicable configuration before you continue with these steps.

2. Expand Advanced and click the [Linked Datastores](#) label.

The [Add Linked Datastore](#) dialog box opens.

3. Select a datastore from the [Select datastore to link to](#) dropdown list.

The datastore that you select is the source datastore that contains the database link that the target datastore uses. The software lists only the datastores with the database types that support linked datastores.

i Note

The target datastore is the one you have opened for editing right now.

❖ Example

You have a target datastore named DS_Emp that contains employee information. You have another datastore named DS_Sales that contains sales information. To associate employee information with sales information, you open the DS_Emp datastore editor and select DS_Sales from the [Select datastore to link to](#) dropdown list.

i Note

The datastore editor allows you to set up only one database link between a target datastore and a source datastore pair. Therefore, if target datastore DS_Emp already has a link to the source DS_Sales, you cannot import another database link that associates DS_Emp with DS_Sales.

- Click **OK** to close the *Add Linked Datastore* dialog box.
The *Linked Datastore* dialog box contains **Not Linked**.
- Select the **Browse** button to the right of the *Linked Datastore* dialog box.
The *Database Link* dialog box opens.
- Select *Use the database link*.

i Note

To remove an existing link, select *Do not link*.

- Select a database link from the *Use the database link* text box dropdown list.
This list contains links that you previously defined for the DBMS.
- Select the source datastore configuration from the *Configuration in Datastore* dropdown list.
- (Optional) Select *Details* to view additional information about the links or to test them.
The *Details* dialog box opens. The checkmark indicates the link to use for testing. Click **OK** to close the *Details* dialog box.
- Click **OK** to close the *Database Link* dialog box.
- Click **OK** to save your changes and to close the datastore editor.

9.4 Adapter datastores

Adapter datastores provide access to application data through an adapter.

Depending on the adapter implementation, adapters allow you to perform the following tasks:

- Browse application metadata
- Import application metadata into the repository
- Move batch and real-time data between SAP Data Services and applications

SAP offers an Adapter Software Development Kit (SDK) to develop your own custom adapters. Also, you can buy the software prepackaged adapters to access application metadata and data in any application. For more information on these products, contact your SAP sales representative.

Adapters are represented in Designer by adapter datastores. Jobs provide batch and real-time data movement between the software and applications through adapter datastore subordinate objects.

Adapter datastore subordinate objects

Subordinate Objects	Use as	Used for
Tables	Source or target	Batch data movement
Documents	Source or target	
Functions	Function call in query	
Message functions	Function call in query	Real-time data movement

Subordinate Objects	Use as	Used for
Outbound messages	Target only	

Adapters can provide access to application data and metadata, or just metadata. For example, if the data source is compatible with SQL, the adapter could be designed to access metadata. The software extracts data from or loads data directly to the application.

For detailed information about installing, configuring, and using adapters, see *Supplement for Adapters*.

Parent topic: [Datastores \[page 92\]](#)

Related Information

[Database datastores \[page 93\]](#)

[Persistent cache datastores \[page 181\]](#)

[Linked datastores \[page 186\]](#)

[Application datastores \[page 191\]](#)

[Web service datastores \[page 194\]](#)

[Change a datastore definition \[page 195\]](#)

[Create and manage multiple datastore configurations \[page 197\]](#)

[Source and target objects \[page 277\]](#)

[Real-time source and target objects \[page 741\]](#)

[Data Services adapters](#)

[Adapters](#)

9.5 Application datastores

Application datastores contain connection information to the specific application.

The following table describes the applications available when you create an application datastore.

Descriptions of application datastores

Application	Datastore
Google BigQuery	Connects to your BigQuery project. • Upload data generated by Data Services to existing tables in your Google BigQuery account. Then run queries. • Extract data from your Google BigQuery tables to use as a source in Data Services data flows. See the <i>Supplement for Google BigQuery</i> for more information.
JDE OneWorld	Connects to the database in your JDE OneWorld application. Data Services bases the configuration options in the datastore based on the database type. The JDE OneWorld datastore works with the following database types: • DB2 • Microsoft SQL Server • ODBC • Oracle For details about configuring the datastore options for JD Edwards applications, refer to the <i>Supplement for J.D. Edwards</i>.
JDE World	Connects to the ODBC database in your JDE World application. For details about configuring the datastore options for JD Edwards applications refer to the <i>Supplement for J.D. Edwards</i>.
Oracle Applications	Connects to data in your Oracle applications. For details about configuring the datastore options for Oracle applications, refer to the <i>Supplement for Oracle Applications</i>.
PeopleSoft	Connects to the following database types in your PeopleSoft application. Options vary based on the database type you select. Applicable PeopleSoft databases are: • Microsoft SQL Server • Oracle For more information about PeopleSoft applications, refer to the <i>Supplement for PeopleSoft</i>.
Replication Server CDC	Connects to changed data capture (CDC) jobs using SAP PowerDesigner Data Movement model.
SAP Applications	Connects to the applicable SAP application. Find descriptions for SAP application datastores in the <i>Supplement for SAP</i>.
SAP BW Source	
SAP BW Target	

Application	Datastore
SAP DQM Microservices	Connects to SAP Data Quality Management, microservices for location data for use of the DQM Microservices transform in Data Services. Find datastore option descriptions for the DQM Microservices datastore in the <i>Supplement for SAP</i>.
Siebel	Connects to the following database types in your Siebel application: • DB2 • Microsoft SQL Server • Oracle For more information about Siebel applications, refer to the <i>Supplement for Siebel</i>.
Web Service REST	Representational State Transfer (REST or RESTful) web service. A design pattern for the World Wide Web. Call a REST server and then browse through and use the data the server returns. Find complete option descriptions in the <i>Integrator Guide</i>.
Web Service SOAP	Connection protocol for XML messages. Invoke real-time services using any of the following methods: • Message Client API • TCP/IP • Proprietary XML using HTTP Find complete option descriptions in the <i>Integrator Guide</i>.

After you create a datastore, import metadata about the objects, such as tables and functions, into that datastore in the object library.

For complete information about adapter datastores, see the *Supplement For Adapters*.

Parent topic: [Datastores \[page 92\]](#)

Related Information

[Database datastores \[page 93\]](#)

[Persistent cache datastores \[page 181\]](#)

[Linked datastores \[page 186\]](#)

[Adapter datastores \[page 190\]](#)

[Web service datastores \[page 194\]](#)

[Change a datastore definition \[page 195\]](#)

[Create and manage multiple datastore configurations \[page 197\]](#)

9.6 Web service datastores

Web service datastores contain connection information to an external web service-based data source.

For more information about accessing web services from Data Services, see the *Integrator Guide*.

Parent topic: [Datastores \[page 92\]](#)

Related Information

[Database datastores \[page 93\]](#)

[Persistent cache datastores \[page 181\]](#)

[Linked datastores \[page 186\]](#)

[Adapter datastores \[page 190\]](#)

[Application datastores \[page 191\]](#)

[Change a datastore definition \[page 195\]](#)

[Create and manage multiple datastore configurations \[page 197\]](#)

9.6.1 Defining a web service datastore

Define at least one datastore for each web service application with which you exchange data.

Ensure that you have the appropriate access privileges to the web service before you create the datastore.

1. Right-click in the [Datastores](#) tab of the object library and select [New](#).
2. Enter the name of the new datastore in the [Datastore name](#) text box.
Use alphanumeric characters and underscores in the name. Do not use blank spaces in the name.
3. Select the [Datastore type](#).
Choose [Web Service REST](#) or [Web Service SOAP](#).
When you select a [Datastore Type](#), Data Services displays other options relevant to that type.
4. Specify the [Web Service URL](#).
Ensure that the URL accepts connections and returns the WSDL. For REST web services, either enter a URL or the path to the local WADL file. Read about web services technologies in the *Integrator Guide*.
5. Click [OK](#) to save the datastore and close the datastore editor.
Data Services saves the datastore configuration in your repository and the new datastore appears in the object library.

9.6.2 Browse WSDL and WADL metadata through a web services datastore

Browse WSDL and WADL metadata using the applicable web services datastore.

Data Services stores metadata information for all imported objects in a datastore. You can also use Data Services to view metadata for nonimported objects and to check whether the metadata has changed for objects already imported.

See "Web services technologies" in the *Integrator Guide* for more information about WSDL and WADL files.

Related Information

[Datastore metadata \[page 102\]](#)

9.7 Change a datastore definition

Change a datastore definition by editing the datastore options or properties.

Datastore options control the operation of the datastore. For example, the options that contain the connection information are datastore options.

Properties contain information about the datastore. For example, the name and description of the datastore and the date on which it was created are datastore properties. Properties do not affect the datastore operation.

[Changing datastore options \[page 196\]](#)

Change datastore options to change the datastore operations, such as how it connects to the data source.

[Changing datastore properties \[page 196\]](#)

For most datastores, you can add or edit the datastore description.

Parent topic: [Datastores \[page 92\]](#)

Related Information

[Database datastores \[page 93\]](#)

[Persistent cache datastores \[page 181\]](#)

[Linked datastores \[page 186\]](#)

[Adapter datastores \[page 190\]](#)

[Application datastores \[page 191\]](#)

[Web service datastores \[page 194\]](#)

9.7.1 Changing datastore options

Change datastore options to change the datastore operations, such as how it connects to the data source.

1. Open the [Datastores](#) tab in the object library.
2. Right-click the applicable datastore name and choose [Edit](#).

The datastore editor opens.

3. Perform the following tasks in the editor:
 - Change one or more of the connection options for the datastore. For example, change the [Database server name](#), [Database name](#), and so on. The options available to edit are based on the datastore type.
 - Change advanced options by clicking [Advanced](#). For example, change the [Rows per commit](#) or the [Additional session parameters](#), and so on.
 - Edit or add additional configurations by clicking [Edit](#) at the bottom of the dialog box. After you edit configurations or add a new one, change the default configuration to use if applicable.
4. Click [OK](#) to save your changes and close the datastore editor.

The changes take effect immediately.

9.7.2 Changing datastore properties

For most datastores, you can add or edit the datastore description.

1. Open the [Datastores](#) tab in the object library.
2. Right-click the applicable datastore and select [Properties](#).

The [Properties](#) dialog box opens.

3. Change or view datastore properties. For example:
 - In the [General](#) tab, add or change the [Description](#).
 - In the [Attributes](#) tab and the [Class Attributes](#) tab, options are view only and you cannot edit them
4. Click [OK](#) to close the [Properties](#) dialog box.

Related Information

[Datastore properties](#)

9.8 Create and manage multiple datastore configurations

Create multiple configurations in a datastore to group connections for similar sources or targets.

Using a datastore with multiple datastore configurations provides greater ease-of-use for job portability scenarios, such as the following:

- Different databases for design and distribution
- Migration to different environments such as DEV, TEST, and PROD
- Databases with different versions or locales
- Databases for central and local repositories

i Note

You can create multiple configuration datastores for any datastore type except for memory datastores.

[Multiple configuration datastore terminology \[page 198\]](#)

When we describe datastores with multiple configurations, we use specific terminology.

[Why use multiple datastore configurations? \[page 199\]](#)

The main reason for using datastores with multiple configurations is to save time.

[Creating a new datastore configuration \[page 199\]](#)

Create a new datastore configuration in an existing datastore using the datastore editor.

[About working with Aliases \[page 202\]](#)

An alias is a logical owner name that you create for objects that you use in different database environments.

[Creating an alias \[page 202\]](#)

Create an alias for a datastore configuration, and use it for multiple configurations.

[Functions to identify the configuration \[page 203\]](#)

SAP Data Services provides functions that are useful when working with multiple source and target datastore configurations.

[Datastore configuration in dataflows \[page 204\]](#)

Changing the datastore configuration in a data flow can modify the data flow language under certain circumstances.

[Portability solutions \[page 206\]](#)

To quickly change connections to a different source or target database, set multiple source or target configurations for a single datastore.

[Rename Owner option \[page 212\]](#)

A shared alias name in a central repository environment makes tracking the object easier when it has been checked out by multiple users.

[Define a system configuration \[page 216\]](#)

When designing jobs, determine content of system configurations based on your business environment and rules.

Parent topic: [Datastores \[page 92\]](#)

Related Information

[Database datastores \[page 93\]](#)

[Persistent cache datastores \[page 181\]](#)

[Linked datastores \[page 186\]](#)

[Adapter datastores \[page 190\]](#)

[Application datastores \[page 191\]](#)

[Web service datastores \[page 194\]](#)

[Change a datastore definition \[page 195\]](#)

9.8.1 Multiple configuration datastore terminology

When we describe datastores with multiple configurations, we use specific terminology.

The following table contains common terms relating to multiple configuration datastores and their definitions.

Term	Definition
Datastore configuration	A set of options that you configure for a datastore. For example, configurable options include a database connection name, database type, user name, password, locale, and so on. A datastore can have more than one set of configurations. Data Services uses the datastore configuration for browsing and importing database objects such as tables and functions, and for executing jobs.
System configuration	Defines a set of datastore and file location configurations to use together when running a job. For convenience, associate substitution parameter configurations to system configurations.
Default datastore configuration	The datastore configuration that Data Services uses when you use the datastore in a job, and when you have not specified a system configuration. If a datastore has more than one configuration, select a configuration as the default. If a datastore has only one configuration, Data Services uses it as the default configuration.
Current datastore configuration	The datastore configuration that Data Services uses to execute a job. Data Services uses the default datastore configuration as the current configuration at job execution time under the following circumstances: • You do not create a system configuration • The system configuration does not specify a configuration for a datastore
Database objects	The tables and functions that you import using the datastore connection information.

Term	Definition
Owner name	The owner name of a database object, such as a table, in an underlying database. Owner name is also known as database owner name or physical owner name. Most database objects have owners. Some database objects do not have owners. For example, database objects in an ODBC datastore connecting to an Access database do not have owners.
Alias	A logical owner name. Use an alias to substitute the logical owner name for the object owner name for the object that you import. If you have different owner names in different database environments, create an alias for each environment. Create alias names in the datastore editor for each configuration.
Dependent objects	Dependent objects are the objects in which the database object is used. Dependent objects can be jobs, work flows, data flows, and custom functions. Find out if an object has dependent objects by right-clicking the object name and selecting View Where Used from the dropdown menu.

9.8.2 Why use multiple datastore configurations?

The main reason for using datastores with multiple configurations is to save time.

Using multiple datastore configurations decreases end-to-end development time in a multi-source, 24x7, enterprise data warehouse environment. Use multiple datastore configurations to decrease the work in porting jobs among different database types, versions, and instances. For example, with multiple configuration datastores, perform porting in a few simple steps:

1. Create a new configuration within an existing source or target datastore.
2. Add a datastore alias and map configurations with different object owner names to the alias.
3. Define a system configuration and add datastore configurations for a particular environment.
4. Select a system configuration when you execute a job.

9.8.3 Creating a new datastore configuration

Create a new datastore configuration in an existing datastore using the datastore editor.

1. From the [Datastores](#) tab of the object library, right-click the applicable datastore and select [Edit](#).

Memory datastores cannot have multiple configurations.

2. Click [Advanced](#) to expand the existing configuration information.

Each datastore has at least one configuration. If only one configuration exists, it is the default configuration.

3. Click [Edit](#) at the bottom of the datastore editor to open the [Configurations for Datastore](#) dialog box.

4.  Click the [Create New Configuration](#) icon on the toolbar.

The [Create New Configuration](#) dialog box opens.

5. Complete the options as described in the following table.

Option	Description
<i>Name</i>	Specifies a name for the configuration. Ensure that you use a unique, logical configuration name.
<i>Database type</i>	Specifies the database type for this configuration. Select a database type from the dropdown menu.
<i>Database version</i>	Specifies the version of the database type you chose. Select a version from the dropdown menu.
<i>Use values from</i>	Specifies the version of the database type that Data Services uses to automatically set target table, Data_Transfer target table, and—or SQL transform values. Data Services uses the following information to either automatically populate this option, or enable the option so that you can select a version. Automatically populated when: - There are other configurations associated with the same database type. - The database version for the new configuration is later than the version in the existing configurations. Populate manually when: - There are no existing configurations associated with the same database type. - There are existing configurations associated with the same database type, but they are for later versions than the version of the new configuration. To populate manually, select a value from the dropdown menu. Options include the following: - Versions from the existing configuration. - Default value for the database type and version you chose.

Option	Description
<i>Restore values if they already exist</i>	Specifies whether Data Services restores values from a deleted configuration that used the same database type and version. Note The software saves all associated target table, Data_Transfer target table, and—or SQL transform values for deleted datastore configurations. - Selected: Data Services uses customized target table, Data_Transfer target table, and—or SQL transform values from previously deleted datastore configurations. Selected is the default. - Deselected: Data Services does not attempt to restore target table, Data_Transfer target table, and—or SQL transform values. You provide new values.

- Click **OK** to save the new configuration.

If there are current Data Services objects that use the database objects from the datastore, such as a data flow, Data Services adds the new database type and version values to the target table, Data_Transfer target table, and—or SQL transform.

To ensure that you know what objects are changed as a result of the new configuration, Data Services displays the *Added New Values - Modified Objects* dialog box. The *Added New Values - Modified Objects* dialog box provides detailed information about affected data flows and modified objects. After you close *Added New Values - Modified Objects*, Data Services displays the *Output* dialog box with the same information.

Data Services requires that you designate one configuration as the default configuration for each new or existing datastore in your repository. The software uses the default configuration to import metadata and also preserves the default configuration during export and multiuser operations.

Data Services designates the first datastore configuration as the default configuration. If you add one or more additional configurations, you can choose a different default configuration by setting the option *Default configuration* to **Yes**.

When you export a repository, Data Services preserves all configurations in all datastores including related target table, Data_Transfer target table, and—or SQL transform settings. If you export a datastore that already exists in the target repository, Data Services overrides configurations in the target repository with the exported datastore configurations. Data Services exports system configurations separate from other job related objects in the repository.

9.8.4 About working with Aliases

An alias is a logical owner name that you create for objects that you use in different database environments.

Create an alias in the [Configurations for Datastore](#) dialog box that you open from the datastore editor. Create an alias for any datastore configuration as applicable. An alias maps to an owner name.

❖ Example

If the real owner name of a table is **real_owner1**, create an alias name of **ALIAS1** and relate the alias name with the real owner name. Each time you use the table object in Data Services, the owner name displays as **ALIAS1**.

When you delete an alias name, the delete operation also deletes each owner name applied for each configuration. The software removes the selected row from the [Configurations for Datastore](#) dialog box, which includes the alias and all assigned owner names.

9.8.5 Creating an alias

Create an alias for a datastore configuration, and use it for multiple configurations.

Edit an existing datastore, or create a new datastore so that you have the datastore editor opened.

1. Click [Advanced](#).
2. Click the option name [Aliases \(Click here to create\)](#).

The [Create New Alias](#) dialog box opens.

3. Enter an alias name in the [Alias Name in Designer](#) text box.

Use only alphanumeric characters and underscores.

4. Click [OK](#).

The [Create New Alias](#) dialog box closes and Data Services displays the alias name under the [Aliases \(Click here to create\)](#) option.

5. Type an owner name in the text box next to the alias name or by selecting an owner name from the dropdown list. Enter or select an owner name for each configuration column.

When you define a datastore alias, the software substitutes your specified datastore configuration alias for the real owner name when you import metadata for database objects. You can also rename datastore tables and functions after you import them.

Related Information

[Renaming table and function owner \[page 213\]](#)

[Imported metadata from database datastores \[page 107\]](#)

9.8.6 Functions to identify the configuration

SAP Data Services provides functions that are useful when working with multiple source and target datastore configurations.

The following table contains descriptions for the functions to use specifically for datastore identification. The functions all belong to the SAP Data Services Miscellaneous category.

Datastore functions

SAP Data Services

Function	Description
db_type	Returns the database type for the current datastore configuration.
db_version	Returns the database version for the current datastore configuration.
db_database_name	Returns the database name for the current datastore configuration when the database type is MS SQL Server or SAP ASE.
db_owner	Returns the real owner name that corresponds to the given alias name under the current datastore configuration.
current_configuration	Returns the name of the datastore configuration that is in use at runtime.
current_system_configuration	Returns the name of the current system configuration. If there is no system configuration defined, returns a NULL value.

Data Services links any target table, Data_Transfer target table, and—or SQL transform settings in a data flow to the related datastore configurations. Enable a SQL transform to perform successfully regardless of the configuration the Job Server uses at execution time by using variable interpolation in SQL text using the datastore functions.

Use the Administrator to select a system configuration when you schedule a job, for example. Also view the underlying datastore configuration. When you perform the following tasks, the Administrator enables you to view the underlying datastore configuration:

- Execute batch jobs
- Schedule batch jobs
- View batch job history
- Create services for real-time jobs

Related Information

[Descriptions of built-in functions](#)

[SQL](#)

[Job portability tips \[page 210\]](#)

[Using the job scheduler](#)

9.8.7 Datastore configuration in dataflows

Changing the datastore configuration in a data flow can modify the data flow language under certain circumstances.

Use multiple configurations successfully by designing your jobs with the same settings for schemas, data types, variables, and so on. Then, when you switch schemas, you do not have to change settings in the data flow.

❁ Example

If you have a datastore with a configuration for Oracle sources and SQL sources, make sure that the table metadata schemas match exactly. Use the same table names, alias names, number and order of columns, as well as the same column names, data types, and content types.

When a datastore has multiple configurations, use caution when you change a configuration option setting. When a data flow contains the following objects, changing the datastore configuration in the data flow can modify the data flow language:

- Table targets
- Table transfer type used in Data_Transfer transform as a target
- SQL transforms

The software adds the table target options and SQL transform text to additional datastore configurations based on the definitions in the existing configuration. The following table describes how the software updates target tables and transforms based on the new configuration settings.

New configuration	Target and transform options
Contains: • Same database type • Version is the same as or newer than an existing configuration	Software automatically uses the values from an existing configuration for values in the SQL transform, Data_Transfer target table, and target table. Values also include bulk loader options.
Contains: • Different database type • Same database type but with an older version than the existing configuration	When the new configuration is for a different database type: Software uses the values from the database type of the new configuration for the target table, Data_Transfer target table, and SQL transform. When the configuration is for the same database type but with an older version than an existing configuration: Software uses values from the existing configuration for the target table, Data_Transfer target table, and SQL transform.

i Note

The software does not copy bulk loader options for targets from one database type to another.

[About the Use Values From option \[page 205\]](#)

SAP Data Services populates values in the data flow based on the setting in the *Use Values From* option in the *Create New Configuration* dialog box.

[Modified Objects report contents \[page 206\]](#)

The *Added New Values-Modified Objects* report dialog box lists the objects that are affected by the new configuration.

Related Information

[Creating a new datastore configuration \[page 199\]](#)

9.8.7.1 About the Use Values From option

SAP Data Services populates values in the data flow based on the setting in the *Use Values From* option in the *Create New Configuration* dialog box.

The values that you set in the *Use values from* dropdown list affect the values that Data Services uses to populate options in the target table, SQL transform, and—or the Data_Transfer target table. For information about the values available in *Use values from*, see [Datastore configuration in dataflows \[page 204\]](#).

The *Use values from* list always contains the following options:

- *Default values*
- Database type and version for each configuration currently associated with the datastore

❖ Example

Your datastore contains two configurations: Oracle 9i and Microsoft SQL Server 2000. The data flow uses a table from the datastore. When you add a new configuration for DB2 to the datastore, the following values appear in the *Use values from* dropdown menu:

- *Default values*
- Oracle 9i
- Microsoft SQL Server 2000

When you select *Default values*, Data Services uses the same defaults that appear for all database targets, Data_Transfer targets, and—or SQL transforms. In SQL transforms, the default SQL text is always blank. The following list contains some of the default target option values:

- *Row commit size* = 1000
- *Column comparison* = Compare by name
- *Delete data from table before loading* = not selected
- *Drop and re-create table* = not selected for regular tables (Selected for template tables)

The *Restore values if they already exist* checkbox in the *Create New Configuration* dialog box is selected by default. If the configuration is for the same database type and version, the software uses values from a deleted configuration. If you deselect the checkbox, you supply the values.

After you click *OK* to save a new configuration, the software performs the following tasks:

- Copies any existing SQL transform, target table, and Data_Transfer target table values.
- Displays a report of the modified objects in a *Modified Object Report* dialog box.

- Displays a report of the modified objects in the [Output](#) dialog box.

9.8.7.2 Modified Objects report contents

The [Added New Values-Modified Objects](#) report dialog box lists the objects that are affected by the new configuration.

Data Services displays the report after you add a new configuration to an existing datastore and click [Apply](#).

Description of the modified object report contents

Report column	Description
Dataflow	Names of the dataflows where the software modified the language.
Modified object	Objects in the dataflow affected by the change.
Object type	Object types affected by the change. For example, table target or SQL transform.
Usage	Usage of the objects, such as source or target.
Has Bulk Loader	Whether the objects have a bulk loader. Contains a Yes or No value.
Bulk Loader Copied	Whether Data Services copied the bulk loader option. Contains a Yes or No value.
Values Existed	Whether there are previous values. Contains a Yes or No value.
Values Restored	Whether Data Services restored the previous values in this dataflow. Contains a Yes or No value.

Use this report as a guide to manually change the values for options of targets, Data_Transfer target tables, and SQL transforms, as needed. In the report, you can sort results by clicking on column headers. You can also save the output to a file.

Data Services also displays the same information in the [Output](#) dialog box after each newly added configuration. Double-click one of the objects in the report to open the data flow in your workspace and view the changes

Related Information

[Datastore configuration in dataflows \[page 204\]](#)

9.8.8 Portability solutions

To quickly change connections to a different source or target database, set multiple source or target configurations for a single datastore.

The multiple configurations make porting datastores from one environment to the next more convenient when you create configurations using the connection configurations for the new environment.

[Migration between environments \[page 207\]](#)

When you move repository metadata to another environment, you use different source and target databases.

[Loading multiple instances \[page 208\]](#)

Loading multiple instances of a data source to a target data warehouse is the same as a migrating between environments except that you use only one repository.

[OEM deployment \[page 208\]](#)

You can design jobs for one database type and deploy those jobs to other database types as an OEM partner.

[Job portability tips \[page 210\]](#)

Job portability tips, and information about specific SAP Data Services features that support portability can make your job porting tasks easier.

[Multiple user development \[page 211\]](#)

When using a central repository management system, multiple developers, each with their own local repository, can check in and check out jobs.

Related Information

[Working in a Multi-user Environment \[page 1010\]](#)

9.8.8.1 Migration between environments

When you move repository metadata to another environment, you use different source and target databases.

The migration process typically includes the following circumstances:

- The two environments use the same database type but may have unique database versions or locales.
- Database objects such as tables and functions may belong to different owners.
- Each environment has a unique database connection name, user name, password, other connection properties, and owner mapping.
- You use a typical repository migration procedure.
 - Use export—import where you export to a file and import the file to a different repository.
 - Use export where you export from one repository to different repository.

The software overwrites datastore configurations that have the same name in the target environment during export. Therefore, add configurations for the target environment to the source repository. For example, add configurations while you are in the development environment before migrating to the test environment. The Data Service export utility saves additional configurations your current repository. After exporting to a new repository, you do not have to edit datastores before running ported jobs in the target environment.

Creating target configurations while you are in the source environment offers the following advantages:

- Minimal production down time: You can start jobs as soon as you export them.
- Minimal security issues: Testers and operators in production do not need permission to modify repository objects.

Related Information

[Administrator Guide: Export/Import](#)

9.8.8.2 Loading multiple instances

Loading multiple instances of a data source to a target data warehouse is the same as migrating between environments except that you use only one repository.

1. Create a datastore that connects to a database.

If the datastore is for an adapter, ensure that the relevant Job Server is running so Data Services can find all available adapter instances for the datastore.

2. In the [Advanced](#) options add an alias and map it to an owner name as applicable.

The software imports datastore objects using an alias name instead of the owner name.

3. Use Data Services renaming feature to rename owners of any existing database objects.

For instructions, see [Renaming table and function owner \[page 213\]](#).

4. Import datastore objects.
5. Create jobs that use the objects.
6. Execute the jobs.
7. Edit the datastore and add additional configurations for each instance of the database.
8. Map owner names from the new database instances to the alias that you defined.
9. Run a job for each database instance.

Related Information

[Migration basics](#)

9.8.8.3 OEM deployment

You can design jobs for one database type and deploy those jobs to other database types as an OEM partner.

The deployment typically has the following characteristics:

- The instances require various source database types and versions.
- Since a datastore can only access one instance at a time, you may need to trigger functions at runtime to match different instances. If this is the case, the software requires different SQL text for functions (such as `lookup_ext` and `sql`) and transforms (such as the SQL transform). The software also requires different settings for the target table (configurable in the target table editor).
- The instances may use different locales.

- Database tables across different databases belong to different owners.
- Each instance has a unique database connection name, user name, password, other connection properties, and owner mappings.
- You export jobs to ATL files for deployment.

[Deploying jobs to other database types as an OEM partner \[page 209\]](#)

Load multiple database instances, use values from existing configurations, and manually update specific values to deploy jobs to other database types as an OEM partner.

9.8.8.3.1 Deploying jobs to other database types as an OEM partner

Load multiple database instances, use values from existing configurations, and manually update specific values to deploy jobs to other database types as an OEM partner.

1. Create and configure jobs for a particular database type following the steps described in [Loading multiple instances \[page 208\]](#).

To support a new instance under a new database type, the software uses the values from previous configurations for populating the values in target table, Data_Transfer target, and—or SQL transform target properties.

2. If you selected a bulk load method for one or more target tables in data flows, and new configurations apply to different database types, manually set the bulk loader options in targets as applicable.

The software does not copy bulk loader options for targets from one database type to another. Therefore, use the Modified Objects report and the list of targets that are automatically set for bulk loading to make manual changes.

3. Manually modify any SQL text for the new database type when the SQL text in the SQL transform is not applicable for the new database type.

If the SQL text contains any hard-coded owner names or database names, replace the names with variables. Variables can supply owner names or database names for multiple database types. Then you do not have to manually modify the SQL text for each environment.

4. If applicable, use the `db_type()` and similar functions to get the database type and version of the current datastore configuration. Then provide the correct SQL text for that database type and version using the variable substitution (interpolation) technique. For a list of functions, see [Functions to identify the configuration \[page 203\]](#).

The software does not support unique SQL text for each database type or version of the `sql()`, `lookup_ext()`, and `pushdown_sql()` functions.

Related Information

[SQL](#)

9.8.8.4 Job portability tips

Job portability tips, and information about specific SAP Data Services features that support portability can make your job porting tasks easier.

The following information can help you understand how Data Services processes jobs that use multiple configuration datastores:

- A datastore can have multiple configurations, each one configured for different database types and versions. However, Data Services considers that the metadata of tables or functions are the same for each configuration in a datastore.

❖ Example

For example, you import a table for a datastore configured for Oracle. Then you use the table in a job that extracts data from DB2. Data Services executes the job as if the table or function metadata is the same.

- Use the metadata that you import with a default configuration of a datastore with all configurations that you define in the same datastore.
- Data Services supports datastore configuration options for some database types or versions that it does not support in others.

❖ Example

Data Services supports parallel reading on Oracle hash-partitioned tables. However, Data Services does not support parallel reading on DB2 or other database hash-partitioned tables. When you set your data flow to run in parallel with an imported Oracle hash-partitioned source table, Data Services reads from each partition in parallel. However, when you run the same job using sources from a DB2 environment, Data Services does not perform parallel reading.

The following Data Services features support job portability:

- Enhanced SQL transform: Enter different SQL text for each different database type and version. Then use variable substitution in the SQL text. The variable substitution allows Data Services to read the correct text for each datastore configuration regardless of database type and version.
- Enhanced target table editor: Configure database table targets for different database types and versions to match their datastore configurations.
- Enhanced datastore editor: Create a new datastore configuration and choose to copy the database properties from an existing configuration or use the current values. Copied database properties include, for example, datastore target options, table target options, and SQL transform text.

The following list contains tips for job portability:

- When you design a job to run from different database types or versions, ensure that you name database tables, functions, and stored procedures the same for all sources. If you create configurations for both case-insensitive databases and case-sensitive databases in the same datastore, name the tables, functions, and stored procedures using all upper-case characters.
- Table schemas should match across the databases in a datastore. Use the same number of columns, same column names, and the same column positions. Ensure that the column data types are the same or compatible. Also define primary and foreign keys the same way.

❖ Example

When you have an Oracle source table with a VARCHAR column, use a VARCHAR column in a Microsoft SQL Server source table. When you have a DATE column in an Oracle source table, use a DATETIME column in the Microsoft SQL Server source table.

- Ensure that stored procedure schemas match. When you import a stored procedure from one datastore configuration and use it for another datastore configuration, Data Services uses the same stored procedure signature for both datastores. Therefore, it is important that the stored procedures match.

❖ Example

When you have a stored procedure for an Oracle function, Data Services requires that you use the same stored procedure function with all other configurations in a datastore. Therefore, Data Services requires that all configurations are for Oracle databases.

If your stored procedure has three parameters in one database configuration, the stored procedure should have exactly three parameters in the other database configurations. Further, the parameters names, positions, data types, in, and out types must match exactly.

9.8.8.5 Multiple user development

When using a central repository management system, multiple developers, each with their own local repository, can check in and check out jobs.

The development environment has the following characteristics:

- It has a central repository and a number of local repositories.
- Data Services merges objects in a multiple development environment through central repository object check in and checkout activities.
- Data Services preserves object history such as versions and labels.
- The database instances have the same database type but may have different versions and locales.
- Database objects may belong to different owners.
- Each database instance has a unique database connection name, user name, password, other connection properties, and owner mappings.

For some of the characteristics, remapping owner names with shared alias names can mitigate potential issues:

- When you import objects with owner names, map owner names with alias names that are shared among all users in the environment. Then, the objects merged through check in and checkout activities in the central repository won't have conflicts.
- To properly preserve the history for all objects in the shared environment, use shared aliases.

9.8.8.5.1 Port jobs in a multiple user environment

Mapping object owner names to shared alias names helps make porting jobs in a multiple user environment easier.

The following list contains information about mapping table owner and function owner names to shared aliases:

- You cannot rename objects in the central repository. Check out the objects to your local repository and use the renaming utility to map the real owner names to the alias names.
- If the objects to be renamed have dependent objects, Data Services displays a message asking you to also check out the dependent objects.
- If you can't check out all the dependent objects, because they are checked out by another user, for example, Data Services displays a message. The message provides you with an option to proceed or cancel the operation. We recommend canceling the check-out process when there are dependent objects that cannot be checked out.
 - If you continue with the check-out process and rename only the objects that you can check out: Data Services does not overwrite the original object in the central repository. Data Services adds the checked-out objects as if they were new objects, and the original object coexists with the object that has the alias name.
 - The dependent data flows that are affected by the renaming process also affect the Usage and Where Used information. Data Services separates the data for the original objects and the new objects.
- If you are able to check out all the dependent objects: Data Services creates new objects and deletes the original objects with the real owner names when you check the objects back into the central repository.
- When you check in objects to the central repository after you remap owner names to aliases, Data Services ensures that you also check in the dependent objects.
- Data Services does not delete original objects from the central repository when you check in the new objects.
- When you check out datastores from the central repository, change them in the local repository, and then check them back into the central repository. The check-in process overrides the datastore configurations in the central repository.
- Avoid overwriting the configurations of other users. When you check out a datastore, create a new configuration instead of changing an existing configuration. Make your configuration the default configuration only when you work in your own environment and your local repository.

When your group completes one phase of development, designate one developer as the last developer. The last developer deletes the datastore configurations that apply to the current environment and keeps or adds the datastore configurations that apply to the next environment.

9.8.9 Rename Owner option

A shared alias name in a central repository environment makes tracking the object easier when it has been checked out by multiple users.

If all users in a multiple user environment keep the alias name when they work with the object in their local repositories, Data Services tracks dependencies for objects that your team checks in and out of the central repository.

You can rename the owner of imported tables, template tables, and functions. When you rename an owner name with an alias name, the instances of a table or function in a data flow are affected. The datastore from which they were imported is not affected.

The software supports both case-sensitive and case-insensitive owner renaming in the following ways:

- If the objects to be renamed are from a case-sensitive database, Data Services preserves case sensitivity.
- If the objects to be renamed are from a datastore that contains both case-sensitive and case-insensitive databases, Data Services bases the casing of the new owner name on the case sensitivity of the default configuration.

To ensure that all objects are portable across all configurations, enter all owner names and object names using uppercase characters.

During the owner renaming process, Data Services:

- Updates the dependent objects to use the new owner name. Dependent objects include jobs, work flows, and data flows that use the renamed object.
- Lists the object in the object library with the new owner name. The object library displays usage or where used information that reflects the number of updated dependent objects.
- Deletes the metadata for the object with the original owner name from the object library and the repository only after it successfully updates all the dependent objects.

[Renaming table and function owner \[page 213\]](#)

Use owner renaming to assign a single metadata alias instead of the real owner name for database objects in the datastore.

[The Rename option in a multiple user environment \[page 214\]](#)

Using the same alias between all local repositories in a multiple user environment enables SAP Data Services to track dependencies for objects that your team checks in and out of the central repository.

9.8.9.1 Renaming table and function owner

Use owner renaming to assign a single metadata alias instead of the real owner name for database objects in the datastore.

Ensure that all instances of the object have the same schema.

1. From the [Datastore](#) tab of the local object library, expand the applicable datastore.
2. Expand the Table, Template Table, or Functions node as applicable.
3. Right-click the applicable table or function and select [Rename Owner](#).

The **Rename Owner for Table** dialog box appears.

4. Enter a name in the [New Owner Name](#) text box.
5. Click [Rename](#).

When you enter a name, Data Services uses it as a metadata alias for the table or function.

A message appears in the [Output](#) dialog box indicating the renaming was successful. If the datastore has two configurations for the object, and the schemas are different but you chose the same name, the software asks that you select a new name for the object.

9.8.9.2 The Rename option in a multiple user environment

Using the same alias between all local repositories in a multiple user environment enables SAP Data Services to track dependencies for objects that your team checks in and out of the central repository.

As you check out and check in objects to the central repository, Data Services behaves in different ways based on the situation. Consider the following questions:

- What is the check-out state of a renamed object?
- Is the object associated with any dependent objects?

The following table describes several scenarios where the answers to these questions are important to know.

Scenario	Behavior
• Object is not checked out. • Object has no dependent objects in the local or central repository.	Data Services renames the object owner.
• Object is checked out. • Object has no dependent objects in the local or central repository.	Data Services renames the object owner.
• Object is checked out. • Object has one or more dependent objects in the local repository.	Data Services displays a message that lists dependent objects that use or refer to the renamed object. • Click Continue: Data Services renames the objects and modifies the dependent objects to refer to the renamed object using the new owner name. • Click Cancel: Data Services returns you to the Rename Owner dialog box. If you rename the object when it is not checked out of the central repository, Data Services leaves the dependent objects in the central repository as is.

Scenario	Behavior
- Object is checked out. - Object has one or more dependent objects.	This scenario contains some complexity. - If you are not connected to the central repository, Data Services displays the following message: "This object is checked out from central repository <X>. Please select Tools Central Repository... to activate that repository before renaming." - Data Services displays a message that lists the dependent objects with a check-out status. If a dependent object is located in the local repository only, the status message indicates that the object is used only in the local repository, and that checkout is not necessary. - If the dependent object is in the central repository but you did not check it out, Data Services displays a message that states the object is not checked out of the central repository. - If you have the dependent object checked out or it is checked out by another user, Data Services displays a message that includes the name of the repository from which the object is checked out. The purpose of this message is to show the dependent objects. In addition, you can choose to check out the necessary dependent objects from the central repository in the message instead of checking the objects out through the central repository object library. The message also has a Refresh List button that updates the checkout status in the list. Refreshing the list is useful when the software identifies a dependent object in the central repository but another user has it checked out. When that user checks in the dependent object, click Refresh List to update the status and verify that the dependent object is no longer checked out. → Tip To use the Rename Owner option to its best advantage, check out associated dependent objects from the central repository. Checking out all dependent objects avoids dependent objects that refer to objects with owner names that do not exist. After you check out the dependent object, Data Services updates the status. If the checkout was successful, the status shows the name of the local repository.

Scenario	Behavior
- Object is checked out. - Object has one or more dependent objects.	When Data Services gives you an option to continue, but you have not checked out one or more dependent objects from the central repository, Data Services displays another dialog box that warns you about objects not yet checked out: - Click No to return to the previous dialog box showing the dependent objects. - Click Yes to proceed with renaming the selected object and to edit the dependent objects. Data Services modifies the object only in the local repository to use the new owner name.  Caution It is your responsibility to maintain consistency between the objects in the central repository and the local repository.
- Object is checked out. - Object has one or more dependent objects.	When Data Services gives you an option to continue, and you have checked out all dependent objects from the central repository, click Continue: - If there are no issues, Data Services renames the owner of the selected object and modifies all dependent objects to refer to the new owner name.  Note It appears as if the original object has a new owner name. However, Data Services did not modify the original object. Data Services creates a new object that is identical to the original object, but uses the new owner name. The original object with the original owner name still exists. Data Services performs an “undo checkout” on the original object. It is your responsibility to check in the renamed object. - Data Services displays the Output dialog box with a rename success message, or a message that the object was not successfully renamed.

9.8.10 Define a system configuration

When designing jobs, determine content of system configurations based on your business environment and rules.

A system configuration consists of datastore configurations. Therefore, your system must have more than one datastore and — or datastores with multiple configurations before you create a system configuration.

Create and edit system configurations in the [Edit System Configurations](#) dialog box. The dialog box contains a list of all existing datastores. It has a toolbar with options to create, duplicate, rename, and delete configurations.

The following table describes the difference between datastore configurations and system configurations.

Datastore configurations	Each datastore defines a connection to a particular data source. Datastores contain source name, server name, user name, and password, for example. Datastores can have more than one configuration. For example, a datastore can have a configuration to connect to tables for your development environment, and another connection to tables in your test environment. If the datastore contains only one configuration, Data Services considers the one configuration as the default configuration. If the datastore contains more than one configuration, you specify the default configuration.
System configurations	Each system configuration contains selected datastore configurations. You select one configuration from applicable datastores to include in your system configuration. For example, you select several configurations that you want to use together when running a job. When you create a system configuration, you can also associate substitution parameter configurations to system configurations. Create multiple system configurations based on your needs.

In many enterprises, a job designer defines the datastore and system configurations and then a system administrator determines the system configuration to use when scheduling or starting a job.

The software maintains system configurations separate from jobs. You cannot check in or check out system configurations from a central repository.

When you upgrade Data Services, or switch environments, export system configurations to a flat file. Then import the flat file to a new environment or software version.

Select the system configuration to use with scheduled jobs, or when you run a job on demand in the [Execution Properties](#) dialog box.

[Creating a system configuration \[page 218\]](#)

Create a system configuration when you have more than one datastore and — or datastores with multiple configurations.

[Creating a system configuration by duplication \[page 218\]](#)

Use a duplicate system configuration as a template to create a new configuration.

[Renaming and deleting system configurations \[page 219\]](#)

The [Edit System Configurations](#) dialog box has toolbar icons for renaming and deleting system configurations.

[Exporting a system configuration \[page 219\]](#)

Export system configurations to use in a new environment or when you upgrade to a newer version of SAP Data Services.

9.8.10.1 Creating a system configuration

Create a system configuration when you have more than one datastore and — or datastores with multiple configurations.

1. In Designer, select **Tools > System Configurations**.

The *Edit System Configurations* dialog box displays.

2. Click the *Create New Configuration* icon  from the tool bar.

The software adds a new column next to the column of existing datastores. The software assigns a default name such as **System_Config_1**.

3. Rename the new system configuration to a name that complies with your naming conventions.

We recommend that you follow a consistent naming convention and use the prefix **sc_** to indicate that the object is a system configuration. Using a naming convention is helpful especially when you export your system configurations.

4. Optional. Select the *Substitution Parameter* dropdown list and select an existing substitution parameter configuration.

The system configuration is associated with the substitution parameter.

5. Select a configuration from each datastore that you want to include in the system configuration.

To exclude a datastore from the system configuration, leave the setting as *<Default configuration>*.

6. Click *OK* to save your system configuration settings and to close the dialog box.

Related Information

[Associating substitution parameter configurations with system configurations \[page 726\]](#)

9.8.10.2 Creating a system configuration by duplication

Use a duplicate system configuration as a template to create a new configuration.

1. In Designer, select **Tools > System Configurations**.

The *Edit System Configurations* dialog box displays. Each existing system configuration appears in a column, with the system configuration name as the column heading.

2. Select the column of the applicable system configuration and click the *Duplicate Configuration* icon  from the tool bar.

The software adds a new column next to the column that you duplicated. The software assigns a name such as *<Existing_SC_Name>_Copy1*.

3. Rename the new system configuration to a name that complies with your naming conventions.

We recommend that you follow a consistent naming convention and use the prefix **sc_** to indicate that the object is a system configuration. Using a naming convention is helpful especially when you export your system configurations.

4. Continue creating the system configuration as in [Creating a system configuration \[page 218\]](#).

9.8.10.3 Renaming and deleting system configurations

The *Edit System Configurations* dialog box has toolbar icons for renaming and deleting system configurations.

Each existing system configuration appears as a column in the *Edit System Configurations* dialog box. The system configuration name appears as the column heading.

1. Select the system configuration column to rename.

The software highlights the entire system configuration column.

2. Click the *Rename Configuration* icon  on the toolbar menu.

The software highlights the name of the system configuration.

3. Enter a new name for the system configuration based on your naming conventions.

We recommend using **sc_** as a prefix for naming system configurations.

Data Services saves the new name after you click *OK*.

4. To delete a system configuration, select the column of the system configuration to delete.

5. Click the *Delete Configuration* icon  from the toolbar menu

The software removes the system configuration column.

6. Click *OK* to save your changes and close the dialog box.

9.8.10.4 Exporting a system configuration

Export system configurations to use in a new environment or when you upgrade to a newer version of SAP Data Services.

1. In the object library, select the *Datastores* tab and right-click inside the tab.
2. Select ► *Repository* ► *Export System Configurations* ► from the dropdown menu.

The *Write Repository Export File* dialog box opens. The software assigns the file name of **repo_export** and selects the ATL file format by default.

3. Rename the file to be exported.

Use a name that you can easily identify when you import the file to the new environment.

4. Select the location for the exported file.
5. Click *Save*.

The software displays the *Output* dialog box with a message indicating that the export completed. The message also contains the number of exported system configurations.

10 Flat file formats

A flat file format is a set of properties that describe the metadata structure of a flat ASCII file.

A flat file format template is a generic description that you can use for multiple data files. You edit the file format using the [File Format Editor](#). Specify options in the [File Format Editor](#) that define specifics such as the following information:

- File type
- Data file information
- File delimiter definitions
- Default formats
- Input and output options
- Locale
- Error handling
- File transfer protocol

Find complete [File Format Editor](#) option descriptions in the *Reference Guide*.

[File format features \[page 222\]](#)

Select several processing features when you configure flat files for processing.

[Creating a new flat file format template \[page 228\]](#)

Create a new flat file format template in SAP Data Services from the object library.

[Replicating file formats \[page 234\]](#)

Quickly create additional file format objects that have the same schemas by replicating and renaming an applicable existing file format.

[Creating a file format from an existing flat table schema \[page 235\]](#)

Steps to use an existing flat file schema in a data flow to quickly create a new flat file format instead of creating a file format from scratch.

[Creating a specific source or target file \[page 235\]](#)

Use a file format template as a source or target in a data flow.

[Editing file formats \[page 236\]](#)

You can modify existing file format templates to match changes in the format or structure of a file.

[File transfer protocols \[page 239\]](#)

Describes two methods to set up file transfer protocols in a file format.

Related Information

[File format](#)

10.1 File format features

Select several processing features when you configure flat files for processing.

Configure flat files to perform the following special tasks:

Feature	Description
Read multiple files at once	Reads and loads multiple files that have the same format and reside in the same location.
Identify the source file name in output	Adds an output field and populates with the name of the source file for each record.
Ignore rows with special markers	Ignores rows that contain specific characters.
Parallel processing	Reads and loads multiple files in parallel for more efficient processing.
Specify error handling processes	Identifies rows that contain data type conversion and row format errors.

[Reading multiple files at one time \[page 223\]](#)

Configure a source format file to read multiple files that have the same format and reside in the same directory.

[Identifying source file names \[page 224\]](#)

Configure your data flow that includes a flat file source and target to output the source file name for each row processed.

[Ignoring rows with specified markers \[page 224\]](#)

Configure the *File Format Editor* so that SAP Data Services ignores rows that contain a specified marker or markers when it reads files.

[About special markers \[page 224\]](#)

Configure a file format so that SAP Data Services ignores rows based on specific markers.

[Setting the number of threads for parallel processing \[page 225\]](#)

SAP Data Services uses parallel threads to read and load files at the same time to maximize performance.

[Error handling for flat file sources \[page 226\]](#)

Configure the *File Format Editor* to report rows in flat file sources that contain specific types of warnings and errors.

Parent topic: [Flat file formats \[page 221\]](#)

Related Information

[Creating a new flat file format template \[page 228\]](#)

[Replicating file formats \[page 234\]](#)

[Creating a file format from an existing flat table schema \[page 235\]](#)

[Creating a specific source or target file \[page 235\]](#)

[Editing file formats \[page 236\]](#)

[File transfer protocols \[page 239\]](#)

10.1.1 Reading multiple files at one time

Configure a source format file to read multiple files that have the same format and reside in the same directory.

Create a data flow that includes the applicable flat file format as the source.

1. Click the source name in the workspace to open the *Source File Editor*.
2. In the *Data File(s)* section in the lower pane, set *Location* to *Local* or *Job Server* as applicable.
3. Specify the location of the *Root directory* by clicking the *Browse* icon, typing the location, or selecting an applicable substitution parameter from the dropdown list.

i Note

Data Services requires that you type the location when the Job Server is on a different computer than Data Services installation. It doesn't matter if you type an absolute path or a relative path as long as the Job Server can access it. Browse is not available in this situation.

4. Enter a list of file names for *File name(s)* in one of the following manners:

- Separate each file name by commas.

❖ Example

`file_1, file_2, file3.`

- Use a wild card character (* or ?) to specify multiple files with a common characteristic in the file name or other metadata.

❖ Example

Enter `*.txt` to specify all files with the `.txt` extension from the specified root directory.

❖ Example

You name your files with the two-character ISO name for country using the format `<xx>data.txt`. Enter the wildcard file name of `??data.txt` to include files such as `jadata.txt` and `dedata.txt`. The wildcard string `??data.txt` does not find the file named `usadata.txt`, however.

5. Continue completing all other applicable options in the source editor.

10.1.2 Identifying source file names

Configure your data flow that includes a flat file source and target to output the source file name for each row processed.

Include a Query transform in your data flow.

Including the source file name in your output results is helpful under the following circumstances:

- You specify multiple files as a source
- You load from different source files on different runs

1. In the *Source Information* group of the *Source File Editor*, set *Include file name* to *Yes*.

Data Services includes a column named DI_FILENAME to contain the name of the source file in the generated output.

2. In the Query transform editor, map the DI_FILENAME column from *Schema In* to *Schema Out*.
3. Complete all other applicable job set up tasks. Execute the job.

Data Services outputs the source file name for each record in the generated output file.

10.1.3 Ignoring rows with specified markers

Configure the *File Format Editor* so that SAP Data Services ignores rows that contain a specified marker or markers when it reads files.

Perform the following steps in the *Source File Editor* or the *File Format Editor*.

1. In the *File Format Editor* dialog box, or in the *Source File Editor*, go to the *Default Format* section.
2. Enter the specific markers in *Ignore row marker(s)*.

For details about how to enter the markers, see [About special markers \[page 224\]](#).

3. If you are in the *File Format Editor*, click *Save & Close* to save your entry. If you are configuring the source in the *Source File Editor*, complete the remaining configuration steps and continue setting the options in the data flow.

When you execute the job, Data Services does not include the rows with the specified markers in the generated output file.

10.1.4 About special markers

Configure a file format so that SAP Data Services ignores rows based on specific markers.

Your data may contain rows prefaced with a marker to indicate a row that contains informational data and not record data. Or your data may have rows that start with a specific string that you want to ignore.

Specify special markers to ignore in either the *File Format Editor* or the *Source File Editor*. To add more than one marker to ignore, enter markers delimited by a semicolon (;). When a marker is also a special character, use the backslash (\) escape character.

The following table provides some examples of how to enter special markers in either the [File Format Editor](#) or the [Source File Editor](#).

Marker value or values	Rows ignored
	None. Blank is the default value. Data Services processes all rows in your data.
abc	Rows that begin with the string abc
abc;def;hi	Rows that begin with abc, def and hi
abc;\;	Rows that begin with abc and ; The backslash character acts as an escape character for the semicolon.
abc;\\;\;	Rows that begin with abc and \ and ; The backslash character acts as an escape character for the backslash and the semicolon.

10.1.5 Setting the number of threads for parallel processing

SAP Data Services uses parallel threads to read and load files at the same time to maximize performance.

1. Right-click the applicable file format in the [Formats](#) tab of the object library and select [Edit](#) from the dropdown menu.

The [File Format Editor](#) opens.

Note

Also perform these steps in the [Source File Editor](#) to enable multiple threading for reading data. Set the option in the [Target File Editor](#) to enable multiple thread loading.

2. Find the [General](#) group of options.
3. Enter an integer in [Parallel process threads](#).

Example

When you have four CPUs on your Job Server computer, enter **4** to use all CPUs to read and load multiple files.

Leave the setting at [{none}](#) to disable parallel processing.

Related Information

[Parallel process threads for flat files](#)

10.1.6 Error handling for flat file sources

Configure the *File Format Editor* to report rows in flat file sources that contain specific types of warnings and errors.

When you configure the *File Format Editor* so that Data Services reports specific warnings and errors during job execution, the software processes rows from flat file sources one at a time. You can view information about the following warnings and errors:

Type	Description
Data type conversion warnings and errors	Error when there is a mismatch between defined data type and actual data type. ❖ Example You define a field type in the <i>File Format Editor</i> as integer, but Data Services encounters varchar data.
Row format warnings and errors	When the defined row format is different than the row format in the data. ❖ Example You select <i>Fixed width</i> for <i>Type</i> in the <i>File Format Editor</i>, but Data Services identifies a row that does not match the stated field width value.
String truncation errors	Error when Data Services has to truncate a string because of format errors. ❖ Example You select <i>Fixed width</i> for <i>Type</i>, but Data Services truncates a row that has more characters than the fixed width size.

Data Services reports file format errors only for flat file sources.

[Error handling option group \[page 226\]](#)

Configure options in the *Error Handling* group in the *File Format Editor* to determine how SAP Data Services reports file format errors.

[About the error file \[page 227\]](#)

Configure SAP Data Services to output errors to a specified file to create an error file by specifying a file name in the *File Format Editor*.

10.1.6.1 Error handling option group

Configure options in the *Error Handling* group in the *File Format Editor* to determine how SAP Data Services reports file format errors.

Configure Data Services to log errors in the following ways:

- Select the types of warnings to log: Data conversion and row format warnings.
- Specify the maximum number of warnings to log

- Select to capture data conversion, row format, and — or string truncation errors
- Specify the maximum number of errors before Data Services stops processing
- Specify whether Data Services writes errors to a file

For complete option descriptions for the *File Format Editor*, see the *Reference Guide*.

10.1.6.2 About the error file

Configure SAP Data Services to output errors to a specified file to create an error file by specifying a file name in the *File Format Editor*.

Data Services outputs the errors in a semicolon-delimited text file. You can have multiple input source files for the error file. The file resides on the same computer as the Job Server.

Entries in an error file have the following syntax:

```
source file path and name; row number in source file; Data Services error;
column number where the error occurred; all columns from the invalid row
```

 **Example**

The following code illustrates a row-format error. We added line breaks for illustration purposes.

```
d:/acl_work/in_test.txt;2;-80104: 1-3-A column delimiter
was seen after column number <3> for row number <2> in file
<d:/acl_work/in_test.txt>. The total number of columns defined is <3>,
so a row delimiter should be seen after column number <3>.
Please check the file for bad data, or redefine the input schema for
the file by editing the file format in the UI.;3;defg;234;def
```

Where	Example data
<source file path and name>	d:/acl_work/in_test.txt
<row number in source file>	2
<Data Services error>	-80104: 1-3-A column delimiter was seen after column number <3> for row number <2> in file <d:/acl_work/in_test.txt>. The total number of columns defined is <3>, so a row delimiter should be seen after column number <3>. Please check the file for bad data, or redefine the input schema for the file by editing the file format in the UI.
<column number where the error occurred>	3
<all columns from the invalid row>	There are three columns delimited by semicolons: defg;234;def

i Note

If you set the file format *Parallel process thread* option to any value greater than 0, or you select *{none}*, Data Services sets the row number in the source file value to -1.

10.2 Creating a new flat file format template

Create a new flat file format template in SAP Data Services from the object library.

1. Open the *Formats* tab of the object library and right-click in a blank area.
2. Select **► New ► File Format ►** from the dropdown menu.

The *File Format Editor* dialog box opens.

3. In the left column under the *General* group, select a flat file type from the *Type* drop list.

i Note

Data Services presents the remaining options based on the type that you choose.

4. Enter a descriptive name in the *Name* text box.

The name you enter is for the file format template. After you save this file format template, you cannot change the name but you can add or edit a description.

5. Complete the remaining options in the properties column as applicable. For option descriptions, see the *Reference Guide*.
6. Click *Save & Close* to save the file format template and close the *File Format Editor*.

[Defining column attributes when creating a file format \[page 229\]](#)

Edit and define the columns or fields in a source file in the columns attributes pane of the *File Format Editor*.

[Number formats \[page 230\]](#)

The dot (.) and the comma (,) are the two most common formats used to determine decimal and thousand separators for numeric data types.

[Date and time formats at the field level \[page 231\]](#)

Override default date and time field formats by defining a format at the field level.

[Use a sample file to define column attributes \[page 232\]](#)

Designate a file location and name in the *File Format Editor* to display sample data on which to base column attributes.

[Defining column attributes using a sample file \[page 233\]](#)

When you use a sample file to define column attributes, SAP Data Services automatically populates many fields based on sample data.

Task overview: [Flat file formats \[page 221\]](#)

Related Information

[File format features \[page 222\]](#)

[Replicating file formats \[page 234\]](#)

[Creating a file format from an existing flat table schema \[page 235\]](#)

[Creating a specific source or target file \[page 235\]](#)

[Editing file formats \[page 236\]](#)

[File transfer protocols \[page 239\]](#)

[File format](#)

10.2.1 Defining column attributes when creating a file format

Edit and define the columns or fields in a source file in the columns attributes pane of the *File Format Editor*.

Create a new or edit an existing file format that does not have the column attributes defined. Use an existing data file as a model when you create the columns. Perform the following steps in the columns attributes pane of the *File Format Editor* dialog box. The columns attributes pane is in the upper right portion of the dialog box.

1. Click *Field1* under the *Field Name* column.

A *Pencil* icon appears at the beginning of the field, and SAP Data Services highlights *Field1* so that you can edit it.

2. Enter a new field name.
3. Click the cell under *Data Type* and choose a data type from the dropdown list.

The dropdown list contains only the data types that Data Services supports.

Data Services automatically populates other columns in the field row based on the data type you chose. For example, when you enter "Name" for *Field Name* and select *varchar* for the *Data Type*, Data Services:

- enters **100** for the *Field Size*
- enters **Name** for the *Content Type*

4. Continue setting up the first field by completing the other applicable attributes in the row.

Additional information about columns:

- *Scale* and *Precision*: Set only for decimal and numeric data types.
- *Content Type*: Software may populate based on the field name or data type. If an appropriate content type is not available, the software defaults *Content Type* to blank.
- *Format*: Optional. Complete for datetime, date, and time fields. Values that you set under *Format* override the default format values that you set in the properties-values column for that data type.

5. Click the empty cell under the first field row to add the second field.
6. Continue adding fields until you complete the schema.
7. Click *Save & Close* to save the file format template and close the *File Format Editor*.

Note

Data Services does not require that you specify columns for a target file for the following reasons:

- The software automatically populates the target schema using the output schema of the object that precedes the target file in the data flow.

- If your target column attributes do not match the output schema of the preceding object, Data Services overwrites your target schema using the output schema.

If you specify the decimal or real data type format of a source column, and the target column name and data type do not match the source schema, Data Services uses the format from the code page of the Job Server computer.

Related Information

[Creating a new flat file format template \[page 228\]](#)

10.2.2 Number formats

The dot (.) and the comma (,) are the two most common formats used to determine decimal and thousand separators for numeric data types.

When you format files in Data Services, data types in which you can use the dot (.) and comma (,) symbols include decimal, numeric, float, and double. Use either symbol for the thousand indicator and either symbol for the decimal separator.

❖ Example

For example: 2,098.65 or 2.089,65.

	Field Name	Data Type	Field Size	Precision	Scale	Format
	Sales Office	int				{none}
	Region	varchar	2			{none}
	Date open	date				#,##0.0
	Country	varchar	7			#,##0,0
*						

The following table lists the options in the [Format](#) dropdown list, and how Data Services processes number formats for each option.

Option	Description
<code>{none}</code>	Data Services expects that the input number contains only the decimal separator. Data Services determines that the decimal separator is either a comma (,) or a dot (.) based on the region set for the Job Server: • Uses a comma (,) as the decimal separator for locales such as Germany and France. These countries consider a comma as the decimal separator. • Uses a dot (.) as the decimal separator for locales such as US, UK, and India. These countries consider a dot as the decimal separator. Data Services loads data to a flat file with either a comma or a dot as the decimal separator based on the same information. Data Services returns an error when an input number contains a thousand separator.
<code>#,##0.0</code>	Data Services expects that the input number contains a comma (,) for the thousand separator and a dot (.) for the decimal separator. Data Services loads data to a flat file with a comma for the thousand separator and a dot for the decimal separator.
<code>#.##0,0</code>	Data Services expects that the input number contains a dot (.) for the thousand separator and a comma (,) for the decimal separator. Data Services loads data to a flat file with a dot for the thousand separator and a comma for the decimal separator.

Data Services also supports numbers that contain leading and trailing decimal signs. For example:
+12,000.00 or 32.32-

10.2.3 Date and time formats at the field level

Override default date and time field formats by defining a format at the field level.

Make both default and field level settings for date and time data types in the [File Format Editor](#).

Definition level	File Format Editor location
Default	Left column. Select or type formats in the Default Format section for Date, Time, and Date-Time.
field	Upper right pane. Select or type formats for any date, datetime, and time data types in the Format column.

❖ Example

You select `yyyy.mm.dd hh24:mi:ss` from the dropdown list for [Date-Time](#) under the [Default Format](#) section at left. Then you type `dd.mm.yyyy hh24:mi:ss` for the [Data Type](#) field of [datetime](#) in the [Format](#) column in

the upper right pane. The software uses the `dd.mm.yyyy hh24:mi:ss` format from the *Format* column in the right pane when it processes datetime data types.

10.2.4 Use a sample file to define column attributes

Designate a file location and name in the *File Format Editor* to display sample data on which to base column attributes.

Keep in mind the following information when you specify a file in the *File Format Editor*:

Purpose	File format settings	Sample file location
You do not intend to use the file format for job execution.	- Set <i>Location</i> to <i>Local</i>. - Browse to set the <i>Root directory</i>. - Browse to set the <i>File name(s)</i> for the sample file.	On the same computer as SAP Data Services.
You intend to use the file format for job execution.	- Set <i>Location</i> to <i>Job Server</i>. i Note Data Services disables the <i>Browse</i> icon. - Type the root directory for <i>Root directory</i>. - Type the file name in <i>File name(s)</i> Use either an absolute path or a relative path to the file, as long as the Job Server can access it. ❖ Example For example, a path on UNIX might be <code>/usr/data/abc.txt</code>. A path on Windows might be <code>C:\DATA\abc.txt</code>.	On the same computer as the Job Server.

Use Telnet to eliminate typing errors

When you manually enter the root directory and file name, you open yourself to errors because of typing mistakes. Also, capitalization is different between Windows and Unix environments.

For Windows, files are not case-sensitive, but they are in UNIX environments. For example, `abc.txt` and `aBc.txt` are different files in the same UNIX directory.

You cannot use Windows Explorer to determine the exact file location on Windows. To reduce the risk of typing errors, establish a connection between the Telnet protocol to the Job Server (UNIX or Windows) computer. Find the full path name of the sample file. Then, copy and paste the path name from the Telnet application directly into the *Root directory* text box in the *File Format Editor*.

10.2.5 Defining column attributes using a sample file

When you use a sample file to define column attributes, SAP Data Services automatically populates many fields based on sample data.

Create a new file format or edit an existing file format. Complete applicable options in the properties column on the left side of the *File Format Editor*. Perform the following steps to select and configure a sample file, then complete the column attributes pane using the sample file data.

1. Under *Data File(s)*, perform the following substeps. For information about the settings to make, see [Use a sample file to define column attributes \[page 232\]](#).
 - a. Select *Local* or *Job Server* for *Location*.
 - b. Browse to or enter a location of the root directory in *Root Directory*.
 - c. Browse for and select the data file, or enter the data file name in *Data File(s)*. Select multiple files if they have the same schema, and they reside in the same location.
2. If the data file is delimited, set the appropriate options in the *Delimiters* section. Choose options from the dropdown lists or specify Unicode delimiters by directly typing the Unicode character code in the text box. Use the format of */xxxx*, where *xxxx* is a decimal Unicode character code. For example, */44* is the Unicode character for the comma (,) character.
3. To use the first row of the data file for field names, set *Skip row header* to *Yes* in the *Input/Output* group.

Data Services displays the column names in the data preview area in the lower right pane, and creates the metadata structure automatically.

4. Optional. Edit the metadata structure as needed in the column attributes area in the upper right pane of the *File Format Editor*.

For example, you can change the field names and edit the metadata structure for both delimited and fixed-width files.

- a. To add a row, or delete a row: Right-click an existing row and select *Insert Above*, *Insert Below*, or *Delete*.
- b. Click a field name to enter a different name for the field.
- c. Select a different data type from the *Data Type* dropdown list.
- d. Update or enter field lengths for the varchar data types. If the file is fixed width, you can also enter a different field length for blob data types.

Note

When you change a data type to blob for fixed-width files, Data Services notifies you that it cannot display blob data in the data preview pane.

- e. Enter values for *Precision* and *Scale* for numeric and decimal data types.
- f. Optional. Enter a value for *Format* when the field is date, time, or datetime. This format information overrides the default format set in the *Default Format* group for that data type.

- g. Optional. Select a type from the [Content Type](#) dropdown list. Data Services may automatically populate [Content Type](#) based on certain field names or data types. Data Services leaves [Content Type](#) blank when it cannot find an applicable content type.
5. Optional. For fixed-width files, edit the metadata structure in the data preview area, which is the lower right pane of the [File Format Editor](#):
 - a. Click to select and highlight columns.
 - b. Right-click to insert or delete fields.
6. Click [Save & Close](#) to save the file format template and close the [File Format Editor](#) dialog box.

10.3 Replicating file formats

Quickly create additional file format objects that have the same schemas by replicating and renaming an applicable existing file format.

1. In the [Formats](#) tab of the object library, right-click an existing file format and choose [Replicate](#) from the dropdown menu.

The [File Format Editor](#) opens, displaying the schema of the copied file format.

2. Type a new, unique name in the [Name](#) option.

! Restriction

Data Services requires that you set a unique name for the replicated file format. Keep in mind that this is the only opportunity you have to name the replicated file format. After you save and close the replicated file format, you cannot modify the name.

3. Edit applicable properties in the left column.
4. To save and not close the new file format schema, click [Save](#) in the lower left corner of the dialog box.

To terminate the replication process, even after you have changed the name and clicked [Save](#), click [Cancel](#) or press [Esc](#) on your keyboard.

5. When you are finished creating the replicated file format, click [Save & Close](#).

Task overview: [Flat file formats \[page 221\]](#)

Related Information

[File format features \[page 222\]](#)

[Creating a new flat file format template \[page 228\]](#)

[Creating a file format from an existing flat table schema \[page 235\]](#)

[Creating a specific source or target file \[page 235\]](#)

[Editing file formats \[page 236\]](#)

[File transfer protocols \[page 239\]](#)

[Reference Guide: File format](#)

10.4 Creating a file format from an existing flat table schema

Steps to use an existing flat file schema in a data flow to quickly create a new flat file format instead of creating a file format from scratch.

Before you follow the steps below, open the applicable data flow that has the object that contains the schema to copy.

1. Click the name of the object.

The object editor opens showing the schema (for example, *Schema In* located in the upper left of the editor).

2. Right-click the schema name and select *Create File format* from the drop menu.

The *File Format Editor* opens showing the current schema settings.

3. Enter a name for the new schema in the *Name* option.

4. Set applicable properties for the new schema as applicable and click *Save & Close*.

The software saves the new file format template in the repository. You can access it from the *Formats* tab of the object library under the *Flat Files* folder.

Task overview: [Flat file formats \[page 221\]](#)

Related Information

[File format features \[page 222\]](#)

[Creating a new flat file format template \[page 228\]](#)

[Replicating file formats \[page 234\]](#)

[Creating a specific source or target file \[page 235\]](#)

[Editing file formats \[page 236\]](#)

[File transfer protocols \[page 239\]](#)

10.5 Creating a specific source or target file

Use a file format template as a source or target in a data flow.

Create a data flow in the work space and select the flat file to use as a source or target.

1. Expand the *Flat Files* folder in the *Formats* tab of the local object library.
2. Drag the file format template to the data flow in your workspace.
3. Select either *Make Source* or *Make Target* from the popup menu based on your intention for the data flow.

The source or target flat file appears in your work space.

4. Click the name of the file format object in the workspace to open the editor.

The object editor opens. The available properties vary based on whether you chose [Make Source](#) or [Make Target](#).

5. Complete the [File name\(s\)](#) and [Location](#) properties under [Data File\(s\)](#).

i Note

For convenience, use variables as file names.

6. Connect the file format object to the applicable object in the data flow based on whether the file is a source or target.

Task overview: [Flat file formats \[page 221\]](#)

Related Information

[File format features \[page 222\]](#)

[Creating a new flat file format template \[page 228\]](#)

[Replicating file formats \[page 234\]](#)

[Creating a file format from an existing flat table schema \[page 235\]](#)

[Editing file formats \[page 236\]](#)

[File transfer protocols \[page 239\]](#)

[Set file names at run-time using variables \[page 718\]](#)

[File format](#)

10.6 Editing file formats

You can modify existing file format templates to match changes in the format or structure of a file.

You cannot change the name of a file format template.

For example, if you have a date field in a source or target file that is formatted as mm/dd/yy and the data for this field changes to the format dd-mm-yy due to changes in the program that generates the source file, you can edit the corresponding file format template and change the date format information.

For specific source or target file formats, you can edit properties that uniquely define that source or target such as the file name and location.

⚠ Caution

If the template is used in other jobs (usage is greater than 0), changes that you make to the template are also made in the files that use the template.

To edit a file format template, do the following:

1. In the object library [Formats](#) tab, double-click an existing flat file format (or right-click and choose [Edit](#)).
The file format editor opens with the existing format values.
2. Edit the values as needed.
Look for properties available when the file format editor is in source mode or target mode.
3. Click [Save](#).

Task overview: [Flat file formats \[page 221\]](#)

Related Information

[File format features \[page 222\]](#)

[Creating a new flat file format template \[page 228\]](#)

[Replicating file formats \[page 234\]](#)

[Creating a file format from an existing flat table schema \[page 235\]](#)

[Creating a specific source or target file \[page 235\]](#)

[File transfer protocols \[page 239\]](#)

[Reference Guide: File format](#)

10.6.1 Editing a source or target file

1. From the workspace, click the name of a source or target file.

The file format editor opens: Either the [Source File Editor](#) or the [Target File Editor](#) opens, displaying the properties for the selected source or target file.

2. Edit the desired properties.

There are some properties that are common for source and target modes. However, there are specific options that are just for source mode or just for target mode. Find complete option descriptions in the [Reference Guide](#).

! Restriction

Any changes you make to values in source or target modes in a data flow override those on the original file format.

To change properties that are not available in source or target mode, you must edit the file's file format template.

3. Click [Save](#).

Related Information

[Reference Guide: Objects, Description of objects, File format, File format properties and modes](#)

10.6.2 Changing multiple column properties

Steps to set the same column properties, such as data type and content type, for multiple columns of a file format in the column attribute area of the [File Format Editor](#).

Use these steps when you are creating a new file format or editing an existing one.

i Note

For clarification, we refer to each field as a separate column. For example, the field First_Name and Last_Name are two columns. We are not referring to the actual columns ([Field Name](#), [Data Type](#), and so on) in the column attribute area.

1. Right-click the name of an existing file format under the [Flat Files](#) folder in the [Formats](#) tab of the object library and select [Edit](#).
The [File Format Editor](#) opens.
2. In the column attributes area (upper right pane) select the multiple columns that you want to change.
 - To choose consecutive columns, select the first column, press the `Shift` key and select the last column.
 - To choose non-consecutive columns, press the `Control` key and click each applicable column.
3. Right click and choose [Properties](#).
The [Multiple Columns Properties](#) window opens. The [Name](#) option shows the name of each column that you selected, and it cannot be changed.
4. Change the applicable options, [Data Type](#) or [Content Type](#) for example, and click [OK](#).
The values that you set in the [Multiple Columns Properties](#) dialog appear as the values in the column attribute area of the [File Format Editor](#).

Related Information

[Flat file formats \[page 221\]](#)

10.7 File transfer protocols

Describes two methods to set up file transfer protocols in a file format.

To use custom settings for file transfer protocol, you can choose to set up a custom transfer program, or you can create a file location object. Both methods need to be set up with a specific file format in the File Format Editor. The following table describes the file transfer protocol methods.

File transfer method	Description
Custom transfer program	For fixed and delimited width file types only. Reads and loads files using the custom transfer program settings in the Custom Transfer group of options in the File Format Editor .
File location object	For a variety of file types. Contains file protocol information for transfer protocols such as FTP, SFTP, or SCP, and includes local and remote server information. Create a file location object and associate it to a file format in the File Format Editor .

Parent topic: [Flat file formats \[page 221\]](#)

Related Information

[File format features \[page 222\]](#)

[Creating a new flat file format template \[page 228\]](#)

[Replicating file formats \[page 234\]](#)

[Creating a file format from an existing flat table schema \[page 235\]](#)

[Creating a specific source or target file \[page 235\]](#)

[Editing file formats \[page 236\]](#)

[Custom file transfers \[page 239\]](#)

[File location objects \[page 256\]](#)

10.7.1 Custom file transfers

The software can read and load files using a third-party file transfer program for flat files.

Set up a custom file transfer program when you create or edit a file format for a fixed or delimited flat file. You can use third-party (custom) transfer programs to:

- Incorporate company-standard file-transfer applications as part of the software job execution
- Provide high flexibility and security for files transferred across a firewall

The custom transfer program option allows you to specify:

- A custom transfer program (invoked during job execution)
- Additional arguments, based on what is available in your program, such as:
 - Connection data
 - Encryption/decryption mechanisms
 - Compression mechanisms

Related Information

[Using a custom transfer program \[page 244\]](#)

10.7.1.1 Custom transfer system variables for flat files

By using variables as custom transfer program arguments, you can collect connection information entered in the software and use that data at run-time with your custom transfer program.

i Note

If you have detailed information like a remote and local server to add to the custom transfer variables, consider creating a file location object instead. The File location dialog provides options to enter server locations.

When you set custom transfer options for external file sources and targets, some transfer information, like the name of the remote server that the file is being transferred to or from, may need to be entered literally as a transfer program argument. You can enter other information using the following system variables:

Data entered for:	Is substituted for this variable if it is defined in the Arguments field
User name	\$AW_USER
Password	\$AW_PASSWORD
Local directory	\$AW_LOCAL_DIR
File(s)	\$AW_FILE_NAME

The following custom transfer options use a Windows command file (Myftp.cmd) with five arguments. Arguments 1 through 4 are system variables:

- User and Password variables are for the external server
- The Local Directory variable is for the location where the transferred files will be stored in the software
- The File Name variable is for the names of the files to be transferred

Argument 5 provides the literal external server name.

i Note

If you do not specify a standard output file (such as `ftp.out` in the example below), the software writes the standard output into the job's trace log.

```
@echo off
set USER=%1
set PASSWORD=%2
set LOCAL_DIR=%3
set FILE_NAME=%4
set LITERAL_HOST_NAME=%5
set INP_FILE=ftp.inp
echo %USER%>%INP_FILE%
echo %PASSWORD%>>%INP_FILE%
echo lcd %LOCAL_DIR%>>%INP_FILE%
echo get %FILE_NAME%>>%INP_FILE%
echo bye>>%INP_FILE%
ftp -s%INP_FILE% %LITERAL_HOST_NAME%>ftp.out
```

Related Information

[File location objects \[page 256\]](#)

10.7.1.2 Custom transfer options for flat files

i Note

If you have detailed information like user names, passwords to add to the custom transfer variables, consider creating a file location object instead. The File location dialog provides options to enter server locations.

Of the custom transfer program options, only the [Program executable](#) option is mandatory.

Entering [User Name](#), [Password](#), and [Arguments](#) values is optional. These options are provided for you to specify arguments that your custom transfer program can process (such as connection data).

You can also use [Arguments](#) to enable or disable your program's built-in features such as encryption/decryption and compression mechanisms. For example, you might design your transfer program so that when you enter `-sSecureTransportOn` or `-CCompressionYES` security or compression is enabled.

i Note

Available arguments depend on what is included in your custom transfer program. See your custom transfer program documentation for a valid argument list.

You can use the [Arguments](#) box to enter a user name and password. However, the software also provides separate [User name](#) and [Password](#) boxes. By entering the `$ <AW_USER>` and `$ <AW_PASSWORD>` variables as [Arguments](#) and then using the [User](#) and [Password](#) boxes to enter literal strings, these extra boxes are useful in two ways:

- You can more easily update users and passwords in the software both when you configure the software to use a transfer program and when you later export the job. For example, when you migrate the job to another environment, you might want to change login information without scrolling through other arguments.
- You can use the mask and encryption properties of the *Password* box. Data entered in the *Password* box is masked in log files and on the screen, stored in the repository, and encrypted by Data Services.

i Note

The software sends password data to the custom transfer program in clear text. If you do not allow clear passwords to be exposed as arguments in command-line executables, then set up your custom program to either:

- Pick up its password from a trusted location.
- Inherit security privileges from the calling program (in this case, the software).

Related Information

[File location objects \[page 256\]](#)

10.7.1.3 Custom transfer options

The custom transfer option allows you to use a third-party program to transfer flat file sources and targets.

You can configure your custom transfer program in the File Format Editor window. Like other file format settings, you can override custom transfer program settings if they are changed for a source or target in a particular data flow. You can also edit the custom transfer option when exporting a file format.

Related Information

[Configuring a custom transfer program in the file format editor \[page 242\]](#)

[File transfer protocols \[page 239\]](#)

10.7.1.3.1 Configuring a custom transfer program in the file format editor

1. Select the *Formats* tab in the object library.
2. Right-click *Flat Files* in the tab and select *New*.

The File Format Editor opens.

3. Select either the *Delimited* or the *Fixed width* file type.

i Note

While the custom transfer program option is not supported by SAP application file types, you can use it as a data transport method for an SAP ABAP data flow.

4. Enter a format name.
5. Select *Yes* for the *Custom transfer program* option.
6. Expand *Custom Transfer* and enter the custom transfer program name and arguments.
7. Complete the other boxes in the file format editor window.

In the Data File(s) section, specify the location of the file in the software.

To specify system variables for Root directory and File(s) in the Arguments box:

- Associate the system variable \$ <AW_LOCAL_DIR> with the local directory argument of your custom transfer program.
- Associate the system variable \$ <AW_FILE_NAME> with the file name argument of your custom transfer program.

For example, enter: -I\$AW_LOCAL_DIR\ \$AW_FILE_NAME

When the program runs, the Root directory and File(s) settings are substituted for these variables and read by the custom transfer program.

i Note

The flag -I used in the example above is a custom program flag. Arguments you can use as custom program arguments in the software depend upon what your custom transfer program expects.

8. Click *Save*.

Related Information

[Supplement for SAP: Custom Transfer method](#)
[Reference Guide: File format](#)

10.7.1.4 Design tips

Use these design tips when using custom transfer options.

- Variables are not supported in file names when invoking a custom transfer program for the file.
- You can only edit custom transfer options in the File Format Editor (or Datastore Editor in the case of SAP application) window before they are exported. You cannot edit updates to file sources and targets at the data flow level when exported. After they are imported, you can adjust custom transfer option settings at the data flow level. They override file format level settings.

When designing a custom transfer program to work with the software, keep in mind that:

- The software expects the called transfer program to return 0 on success and non-zero on failure.
- The software provides trace information before and after the custom transfer program executes. The full transfer program and its arguments with masked password (if any) is written in the trace log. When "Completed Custom transfer" appears in the trace log, the custom transfer program has ended.
- If the custom transfer program finishes successfully (the return code = 0), the software checks the following:
 - For an ABAP data flow, if the transport file does not exist in the local directory, it throws an error and the software stops.
 - For a file source, if the file or files to be read by the software do not exist in the local directory, the software writes a warning message into the trace log.
- If the custom transfer program throws an error or its execution fails (return code is not 0), then the software produces an error with return code and `stdout/stderr` output.
- If the custom transfer program succeeds but produces standard output, the software issues a warning, logs the first 1,000 bytes of the output produced, and continues processing.
- The custom transfer program designer must provide valid option arguments to ensure that files are transferred to and from the local directory (specified in the software). This might require that the remote file and directory name be specified as arguments and then sent to the Designer interface using system variables.

Related Information

[Supplement for SAP: Custom Transfer method](#)

10.7.2 Using a custom transfer program

A custom transfer program refers to a specific third-party file transfer protocol program that you use to read and load flat files. This option applies to fixed width or delimited flat files.

For other flat file types, and for nested schemas, you can set up a file transfer protocol definition by creating a file location object instead of using the custom transfer program settings.

If you select *Yes* for the *Custom Transfer Program* option, you must complete the options in the *Custom Transfer* section later in the properties-values section.

1. Set *Custom Transfer Program* option to *Yes*.
2. Expand the *Custom Transfer* section and enter the custom transfer program name and arguments.

Related Information

[Creating a new flat file format template \[page 228\]](#)

[File transfer protocols \[page 239\]](#)

[File location objects \[page 256\]](#)

11 HDFS file format

The file format for the Hadoop distributed file system (HDFS) describes the file system structure.

Characteristic	Description
Class	Reusable
Access	In the object library, click the Formats tab.
Description	An HDFS file format describes the structure of a Hadoop distributed file system. Store templates for HDFS file formats in the object library. The format consists of multiple properties that you set in the file format editor. Available properties vary by the mode of the editor. The HDFS file format editor includes most of the regular file format editor options plus options that are unique to HDFS.

Related Information

[Flat file formats \[page 221\]](#)

11.1 Configuring custom Pig script results as source

Use an HDFS file format and a custom Pig script to use the results of the PIG script as a source in a data flow.

Create a new HDFS file format or edit an existing one. Use the Pig section of the HDFS file format to create or locate a custom Pig script that outputs data.

Follow these steps to use the results of a custom Pig script in your HDFS file format as a source:

1. In the HDFS file format editor, select [Delimited](#) for [Type](#) in the [General](#) section.
2. Enter the location for the custom Pig script results output file in [Root directory](#) in the [Data File\(s\)](#) section.
3. Enter the name of the file to contain the results of the custom Pig script in [File name\(s\)](#).
4. In the [Pig](#) section, set [Custom Pig script](#) to the path of the custom Pig script. The location must be on the machine that contains the Data Services Job Server.
5. Complete the applicable output schema options for the custom Pig script.
6. Set the delimiters for the output file in the [Delimiters](#) section.
7. Save the file format.

Use the file format as a source in a data flow. When the software runs the custom Pig script in the HDFS file format, the software uses the script results as source data in the job.

Related Information

[HDFS file format objects](#)

12 Working with COBOL copybook file formats

A COBOL copybook file format describes the structure defined in a COBOL copybook file (usually denoted with a .cpy extension).

When creating a COBOL copybook format, you can:

- Create just the format, then configure the source after you add the format to a data flow, or
- Create the format and associate it with a data file at the same time.

This section also describes how to:

- Create rules to identify which records represent which schemas using a field ID option.
- Identify the field that contains the length of the schema's record using a record length field option.

Related Information

[Reference Guide: COBOL copybook source options](#)

[Reference Guide: Data Types, Conversion to or from internal data types \[page 956\]](#)

[File location objects \[page 256\]](#)

[COBOL copybook file format](#)

12.1 Creating a new COBOL copybook file format

1. In the local object library, click the [Formats](#) tab, right-click [COBOL copybooks](#), and click [New](#).

The Import COBOL copybook window opens.

2. Name the format by typing a name in the [Format name](#) field.
3. On the [Format](#) tab for [File name](#), specify the COBOL copybook file format to import, which usually has the extension .cpy.

During design, you can specify a file in one of the following ways:

- For a file located on the computer where the Designer runs, you can use the [Browse](#) button.
- For a file located on the computer where the Job Server runs, you must type the path to the file. You can type an absolute path or a relative path, but the Job Server must be able to access it.

4. Click [OK](#).

The software adds the COBOL copybook to the object library.

5. The [COBOL Copybook schema name\(s\)](#) dialog box displays. If desired, select or double-click a schema name to rename it.

6. Click [OK](#).

When you later add the format to a data flow, you can use the options in the source editor to define the source.

Related Information

[Reference Guide: COBOL copybook source options](#)

12.2 Creating a new COBOL copybook file format and a data file

1. In the local object library, click the [Formats](#) tab, right-click [COBOL copybooks](#), and click [New](#).

The Import COBOL copybook window opens.

2. Name the format by typing a name in the [Format name](#) field.
3. On the [Format](#) tab for [File name](#), specify to the COBOL copybook file format to import, which usually has the extension .cpy.

During design, you can specify a file in one of the following ways:

- For a file located on the computer where the Designer runs, you can use the Browse button.
- For a file located on the computer where the Job Server runs, you must type the path to the file. You can type an absolute path or a relative path, but the Job Server must be able to access it.

4. Click the [Data File](#) tab.
5. For [Directory](#), type or browse to the directory that contains the COBOL copybook data file to import.

If you include a directory path here, then enter only the file name in the Name field.

6. Specify the COBOL copybook data file [Name](#).

If you leave Directory blank, then type a full path and file name here.

During design, you can specify a file in one of the following ways:

- For a file located on the computer where the Designer runs, you can use the Browse button.
- For a file located on the computer where the Job Server runs, you must type the path to the file. You can type an absolute path or a relative path, but the Job Server must be able to access it.

7. If the data file is not on the same computer as the Job Server, click the [Data Access](#) tab. Select [FTP](#) or [Custom](#) and enter the criteria for accessing the data file.
8. Click [OK](#).
9. The [COBOL Copybook schema name\(s\)](#) dialog box displays. If desired, select or double-click a schema name to rename it.
10. Click [OK](#).

The Field ID tab allows you to create rules for indentifying which records represent which schemas.

Related Information

[COBOL copybook file format](#)

12.3 Creating rules to identify which records represent which schemas

1. In the local object library, click the [Formats](#) tab, right-click [COBOL copybooks](#), and click [Edit](#).

The Edit COBOL Copybook window opens.

2. In the top pane, select a field to represent the schema.
3. Click the [Field ID](#) tab.
4. On the Field ID tab, select the check box [Use field <schema name>.field name> as ID](#).
5. Click [Insert below](#) to add an editable value to the Values list.
6. Type a value for the field.
7. Continue (adding) inserting values as necessary.
8. Select additional fields and insert values as necessary.
9. Click [OK](#).

12.4 Identifying the field that contains the length of the schema's record

1. In the local object library, click the [Formats](#) tab, right-click [COBOL copybooks](#), and click [Edit](#).

The Edit COBOL Copybook window opens.

2. Click the [Record Length Field](#) tab.
3. For the schema to edit, click in its Record Length Field column to enable a drop-down menu.
4. Select the field (one per schema) that contains the record's length.

The offset value automatically changes to the default of 4; however, you can change it to any other numeric value. The offset is the value that results in the total record length when added to the value in the Record length field.

5. Click [OK](#).

13 Creating Microsoft Excel workbook file formats on UNIX platforms

Describes how to use a Microsoft Excel workbook as a source with a Job Server on a UNIX platform.

To create Microsoft Excel workbook file formats on Windows, refer to the *Reference Guide*.

To access the workbook, you must create and configure an adapter instance in the Administrator. The following procedure provides an overview of the configuration process. For details about creating adapters, refer to the *Management Console Guide*.

Also consider the following requirements:

- To import the workbook, it must be available on a Windows file system. You can later change the location of the actual file to use for processing in the Excel workbook file format source editor. See the *Reference Guide*.
- To reimport or view data in the Designer, the file must be available on Windows.
- Entries in the error log file might be represented numerically for the date and time fields. Additionally, Data Services writes the records with errors to the output (in Windows, these records are ignored).

Related Information

[Reference Guide: Excel workbook format](#)
[Management Console Guide: Adapters](#)
[Reference Guide: Excel workbook source options](#)
[File location objects \[page 256\]](#)

13.1 Creating a Microsoft Excel workbook file format on UNIX

1. Using the Server Manager (<[LINK_DIR](#)>/bin/svrcfgInstallation Guide for UNIX), ensure the UNIX Job Server can support adapters. See the .
2. Ensure a repository associated with the Job Server is registered in the Central Management Console (CMC). To register a repository in the CMC, see the *Administrator Guide*.
3. In the Administrator, add an adapter to access Excel workbooks. See the *Management Console Guide*.

You can only configure one Excel adapter per Job Server. Use the following options:

- On the Status tab, click the job server adapter at right to configure.
-), ensure theOn the Adapter Configuration tab of Adapter Instances page, click [Add](#).
- On the Adapter Configuration tab, enter the [Adapter instance name](#). Type BOExcelAdapter (required and case sensitive).

You may leave all other options at their default values except when processing files larger than 1 MB. In that case, change the Additional Java Launcher Options value to `-Xms64m -Xmx512m` or `-Xms128m -Xmx1024m` (the default is `-Xms64m -Xmx256m`). Note that Java memory management can prevent processing very large files (or many smaller files).

Note

Starting with Data Services 4.2 SP5, the Excel adapter uses a SAX event-based mechanism to parse the Excel files. If you want to revert it to the way it was processed in version 4.2 SP4 or earlier, set the following flag to false and restart the Job Server. The flag is specified at the Java launcher options for the BOExcelAdapter configuration settings in Management Console.

`DuseSAXEventModelForXlsx=false`

If the `DuseSAXEventModelForXlsx` flag is not specified, it defaults to true and uses the SAX event model.

When processing large Excel files, it is recommended to increase the Java heap size in the Java Launcher Options. For example:

`-Xms256m -Xmx4096m`

4. From the **Administrator** > **Adapter** > **Adapter Instance Status** tab, start the adapter.
5. In the Designer on the **Formats** tab of the object library, create the file format by importing the Excel workbook. For details, see the *Reference Guide*.

Related Information

[Administrator Guide: Registering a repository in the CMC](#)

[Management Console Guide: Adding and configuring adapter instances](#)

[Reference Guide: Excel workbook format](#)

14 Creating Web log file formats

Web logs are flat files generated by Web servers and are used for business intelligence.

Web logs typically track details of Web site hits such as:

- Client domain names or IP addresses
- User names
- Timestamps
- Requested action (might include search string)
- Bytes transferred
- Referred address
- Cookie ID

Web logs use a common file format and an extended common file format.

Common Web log format:

```
151.99.190.27 - - [01/Jan/1997:13:06:51 -0600]  
"GET /~bacuslab HTTP/1.0" 301 -4
```

Extended common Web log format:

```
saturn5.cun.com - - [25/JUN/1998:11:19:58 -0500]  
"GET /wew/js/mouseover.html HTTP/1.0" 200 1936  
"http://av.yahoo.com/bin/query?p=mouse+over+javascript+source+code&hc=0"  
"Mozilla/4.02 [en] (x11; U; SunOS 5.6 sun4m)"
```

The software supports both common and extended common Web log formats as sources. The file format editor also supports the following:

- Dash as NULL indicator
- Time zone in date-time, e.g. 01/Jan/1997:13:06:51 -0600

The software includes several functions for processing Web log data:

- Word_ext function
- Concat_date_time function
- WL_GetKeyValue function

Related Information

[Word_ext function \[page 253\]](#)

[Concat_date_time function \[page 253\]](#)

[WL_GetKeyValue function \[page 254\]](#)

14.1 Word_ext function

The `word_ext` is a string function that extends the `word` function by returning the word identified by its position in a delimited string.

This function is useful for parsing URLs or file names.

Format

```
word_ext(string, word_number, separator(s))
```

A negative word number means count from right to left

Examples

```
word_ext('www.bodi.com', 2, '.') returns 'bodi'.
```

```
word_ext('www.cs.wisc.edu', -2, '.') returns 'wisc'.
```

```
word_ext('www.cs.wisc.edu', 5, '.') returns NULL.
```

```
word_ext('aaa+=bbb+=ccc+zz=dd', 4, '+=') returns 'zz'. If 2 separators are specified (+=), the function looks for either one.
```

```
word_ext(',,,,,aaa,,,,bb,,,c ', 2, '.') returns 'bb'. This function skips consecutive delimiters.
```

14.2 Concat_date_time function

The `concat_date_time` is a date function that returns a datetime from separate date and time inputs.

Format

```
concat_date_time(date, time)
```

Example

```
concat_date_time(MS40."date",MS40."time")
```

14.3 WL_GetKeyValue function

The `WL_GetKeyValue` is a custom function (written in the Scripting Language) that returns the value of a given keyword.

It is useful for parsing search strings.

Format

```
WL_GetKeyValue(string, keyword)
```

Example

A search in Google for `bodi B2B` is recorded in a Web log as:

```
GET "http://www.google.com/search?hl=en&lr=&safe=off&q=bodi+B2B&btnG=Google+Search"
WL_GetKeyValue('http://www.google.com/search?hl=en&lr=&safe=off&q=bodi+B2B&btnG=Google+Search','q') returns 'bodi+B2B'.
```

15 Unstructured file formats

Unstructured file formats are a type of flat file format.

To read files that contain unstructured content, create a file format as a source that reads one or more files from a directory. At runtime, the source object in the data flow produces one row per file and contains a reference to each file to access its content. In the data flow, you can use a Text Data Processing transform such as Entity Extraction to process unstructured text or employ another transform to manipulate the data.

The unstructured file format types include:

Unstructured file format types	Description
Unstructured text	Use this format to process a directory of text-based files including • Text • HTML • XML Data Services stores each file's content using the long data type.
Unstructured binary	Use this format to read binary documents. Data Services stores each file's content using the blob data type. • You can process a variety of document formats by obtaining your input from a variety of binary-format files, then passing that blob to the Text Data Processing transform. In this manner, the following formats can be accepted: ◦ Microsoft Word: 2003, 2007, and 2010 (Office Open XML) ◦ Microsoft PowerPoint: 2003, 2007, and 2010 ◦ Microsoft Excel: 2003, 2007, and 2010 ◦ Adobe PDF: 1.3 – 1.7 ◦ Microsoft RTF: 1.8 and 1.9.1 ◦ Microsoft Outlook E-mail Message: 2003, 2007, 2010 ◦ Generic E-mail Message: ".eml" files ◦ Open Document Text, Spreadsheet, and Presentation: 1.0, 1.1, 1.2 ◦ Corel WordPerfect: 6.0 (1993) – X5 (2010) • You could also use the unstructured binary file format to move a directory of graphic files on disk into a database table. Suppose you want to associate employee photos with the corresponding employee data that is stored in a database. The data flow would include the unstructured binary file format source, a Query transform that associates the employee photo with the employee data using the employee's ID number for example, and the database target table.

Related Information

[Reference Guide: Objects, File format Entity Extraction transform](#)

16 File location objects

File location objects are reusable objects that define properties for a specific file transfer protocol.

The software supports the following file transfer protocols:

- FTP
- SFTP
- SCP
- Local
- Google Cloud Storage
- Azure Cloud Storage
- Azure Data Lake Store
- Amazon S3 Cloud Storage
- Hadoop Distributed file System (HDFS)
- WebHDFS

When you create a file location object, enter file transfer protocol specifics and define a local and remote server.

The software stores file location objects under the File Locations node in the *Format* tab of the local object library.

After you create a file location object, associate it to a format object when you create a new or edit an existing object. To use the transfer protocol information in a data flow, associate the file location object with a source or target in a data flow.

Applicable format objects include flat files, nested schema files (XML, DTD, and JSON), COBOL copybooks, and Excel workbooks.

i Note

Excel workbooks and COBOL copybooks paired with a file location object can only be used as sources in a data flow.

The software uses the remote and local server information and the file transfer protocols in the data flow to move data between the local and remote server.

❖ Example

- Use the data in the local server as a source in a data flow.
- Load generated output data from the data flow into the local server and then to the remote server.

The software provides options to control what happens to any existing data when it transfers the data into a file from remote or local servers. Select options to append, overwrite, delete, or save data in the local or remote servers. These “append” options are available based on the format object type, protocol type, and use type (source or target). If a data file does not already exist in the local or remote server, the software creates the file.

Related Information

[Creating a file location object \[page 259\]](#)

[Manage local and remote data files \[page 257\]](#)

[Reference Guide: Objects, Description of objects, File location objects](#)

[Reference Guide: Objects, Description of objects, File formats](#)

16.1 Manage local and remote data files

SAP Data Services provides options to control what happens to any existing data when it transfers generated data into a file from remote or local servers.

Append options are located in various editor dialogs based on whether you are working with a source or target, and whether you are working with a flat file, nested schema, COBOL copybook, or Excel workbook. Even though some of the options save or delete existing data, we still refer to them as “append” options. The following table lists the append options and how they apply to the local and remote files for sources and targets.

Append file options

Option	File Location Object Used as	Editor Name	Protocol	Description
<i>Delete and re-create file</i>	Target (nested schema)	<i>Nested Schema Target File Editor</i>	FTP, SFTP	Enabled: Overwrites existing remote file with generated output data. Disabled: Appends generated output data from local file to re- mote file.

i Note
Either enabled or disabled: When a remote file does not exist, the software creates the remote file and populates it with the generated output.

Option	File Location Object Used as	Editor Name	Protocol	Description
			SCP	Enabled or disabled: Overwrites the remote file with the contents of the generated output file.
			Local, AWS S3	Enabled or disabled: Appends, overwrites, or creates a local file.
Delete and re-create file	Target (flat files, XML, XML template, JSON, JSON template, and Copybook)	Target File Editor	Azure Data Lake Store	Enabled: Creates or overwrites the existing file in Azure Data Lake Store with the generated data. Disabled: Creates or appends generated data to the existing file.
Delete file after transfer	Source (all applicable format objects)	Nested Schema Format Editor (XML, DTD, or JSON)	Local, AWS S3	Enabled: Deletes the local file after it reads the data into the data flow. Disabled: Saves the local file after it reads the data into the data flow.
Delete file after transfer	Source (flat files, XML, XML template, JSON, JSON template Excel Workbook, and Copybook)	Source File Editor	Azure Data Lake Store	Enabled: Deletes local file after it reads the data into the data flow. Disabled: Saves local file after it reads the data into the data flow.
Delete file after transfer	Target (all applicable format objects)	Target File Editor	Azure Data Lake Store	Enabled: Deletes the local file after it loads data to the target Azure Data Lake Store. Disabled: Saves the local file after it loads data to the target Azure Data Lake Store.

Option	File Location Object Used as	Editor Name	Protocol	Description
Delete file	Target (flat files only)	Target File Editor	FTP, SFTP	Enabled: Overwrites the contents of the remote file with the generated output. Disabled: Appends the generated output to the existing contents of the remote file. Note Enabled or disabled: When a remote file does not exist, the software creates the remote file and populates it with the generated output.
			SCP	Enabled or disabled: Overwrites an existing remote file or creates a new remote file when one does not exist.
			Local, AWS S3	Enabled or disabled: Overwrites or appends data to the local file.

Related Information

[Associate file location objects to file formats \[page 263\]](#)

16.2 Creating a file location object

Create a file location object to specify file transfer protocol and to set local and remote server locations for source and target files.

1. Right-click the File Locations node in the [Formats](#) tab of the object library and select [New](#).

2. Enter a name for the new object in the [Create New File Location](#) dialog box.
3. Select a protocol from the [Protocol](#) dropdown list. The default value is [Local](#).

The [Protocol](#) option setting determines the remaining options. The following table lists brief descriptions for the options. For complete descriptions and other important information, see the [Reference Guide](#).

Option	Brief description
Host	Specifies the remote server name.
Port	Specifies the remote server port number.
Hostkey Fingerprint	Specifies the code for the host computer (SFTP and SCP only).
Authorization Type	Specifies whether you use a password or public key for authorization.
User	Specifies the user name for the specified remote server.
Password	Specifies the password related to the user for the remote server.
SSH Authorization Private Key File Path	Specifies the path to the private key file (SFTP and SCP only).
SSH Authorization Private Key Passphrase	Specifies the passphrase related to the specified private key file (SFTP and SCP only).
SSH Authorization Public Key File Path	Specifies the path to the public key file (SFTP and SCP only).
Connection Retry Count	Specifies the number of times Data Services should try to connect to the server.
Connection Retry Interval	Specifies the time in seconds between which Data Services waits to retry connecting to the server.
Remote Directory	Optional. Specifies the file path to the remote server.
Local Directory	Specifies the path to the local server.

4. When you are finished setting applicable options, click [OK](#).

The software creates the file location object.

After you create the file location, optionally create additional configurations in the file location editor. To add additional configurations, click [Edit](#) at the bottom of the file location editor to open the [Create New Configuration](#) dialog box.

Related Information

[Reference Guide: Objects, Description of objects, File location objects, Common options](#)

[Associate file location objects to file formats \[page 263\]](#)

[Create multiple configurations of a file location object \[page 262\]](#)

16.3 Obtaining SSH authorization

When you choose *Public Key* for *Authorization Type*, obtain SSH authorization to complete the SSH options in the *Connection* group of the file location editor.

The process to obtain SSH authorization varies based on your host system.

1. Use your Windows certificate tool to generate a private and public key pair. For Linux, execute the following command:

```
~/ .ssh
```

2. Open the `authorized_keys` file. For linux remote host, make `~/ .ssh` your current directory.
3. Append the content of the public key file to the `authorized_keys` file.

If you use RSA Data Security encryption, the Linux command generates the `id_rsa` private key file and `id_rsa.pub` public key file.

4. Copy the private key file and the public key file to the Job Server host.

If you use RSA Data Security encryption in Linux, copy the private key file `id_rsa` and the public key file `id_rsa.pub` to the Job Server host.

5. Enter the private key file name and the public key file name to the options in the file location object. If you use RSA Data Security encryption in Linux, you would enter the following values to complete the file location object options:
 - *SSH Authorized Private Key File Path* = `id_rsa`
 - *SSH Authorized Public Key File Path* = `id_rsa.pub`
 - *SSH Authorized Private Key Passphrase* = Enter a passphrase only if you defined a passphrase when you generated the SSH private key.

16.4 Editing an existing file location object

Edit a file location object to change settings or to add additional configurations.

1. Right-click the name of the *File Locations* object to edit in the *Formats* tab of the object library and select *Edit*.

The *Edit File Location <filename>* dialog box opens.

2. Edit the applicable options and click *OK*.

You name the file location object when you create it. You cannot change the name of the file location object when you edit it.

Related Information

[Creating multiple file location object configurations \[page 262\]](#)

[Reference Guide: Objects, Description of objects, File location objects, Common options](#)

16.5 Create multiple configurations of a file location object

Create multiple file location object configurations for each type of file transfer protocol you use.

Each file location object has at least one configuration. If only one configuration exists, it is the default configuration. When you create multiple configurations in one file location object, you can change the default configuration to any of the other configurations in the object.

Using multiple configurations is convenient. For example, when you use different protocols based on your environment, you can create one file location object that contains configurations for your test and production environments. In your test environment you use a local protocol because testing is performed in-house. When you migrate to production, you use SFTP to securely transfer your files between a remote and local server. Switch protocols by editing the file location object that you have used in your job to use a different default configuration.

16.5.1 Creating multiple file location object configurations

Steps to create additional file location configurations in a single file location object, and select a default configuration.

1. Right-click an existing file location object name under *File locations* in the *Format* tab in object library and click *Edit*.

The *Edit File Location <filename>* dialog box opens.

2. Click *Edit...*

The *Configurations for File Location <filename>* dialog box opens.

3. Click the create new configuration icon ()

The *Create New Configuration* dialog opens.

4. Enter a unique name for the configuration and click *OK*.

A new configurations column appears with the new name as the heading.

5. Select *Yes* for *Default configuration* if this configuration is your default.

i Note

If you have multiple configuration in the file location object, the software automatically switches *Default configuration* for the other configurations to *No*.

6. Select a protocol from the *Protocol* drop menu.

The remaining options that appear after the *Protocols* option are based on the protocol type you choose. However, if you have multiple configurations (more than one column of settings), there may be options listed that don't apply to the protocol you are defining. Options that don't apply to a configuration appear with "N/A" in the column.

7. Complete the other applicable options and click *OK*.

For option descriptions, see the *Reference Guide*.

Related Information

[Reference Guide: Objects, Description of objects, File location objects, Common options](#)

16.6 Associate file location objects to file formats

To use the file transfer protocol information that you entered into a file location object, you associate the file location object with a specific file source or target in a data flow.

You can also include a file location object when you create or edit a new format.

i Note

Some formats are not applicable for file location objects. In those cases, the file location object options don't apply.

The following table provides an overview of how to associate a specific file location object with a specific format type.

Format type	Associate file location object
Flat file	Create or edit a flat file format. Add the file location object information in the <i>Data Files</i> section of the <i>File Format Editor</i> .
DTD, XML, or JSON schema file	Edit a schema file. Add the file location object information in the <i>Format</i> tab of the <i>Format Editor</i> .
COBOL copybook	Edit a COBOL copybook. Add the file location information in the <i>Data File</i> tab of the <i>Edit COBOL Copybook</i> dialog.

Format type	Associate file location object
Excel workbook	Edit an Excel workbook. Add the file location information in the <i>Format</i> tab of the <i>Import Excel Workbook</i> dialog.

Related Information

Designer Guide: File location objects, Manage local and remote data files [page 257]

Designer Guide: File locations, Associating flat file format with file location object [page 264]

Designer Guide: File locations, Associating XML, DTD, or JSON schema files [page 265]

Designer Guide: File locations, Associating COBOL copybooks with a file location object [page 266]

Designer Guide: File locations, Associating Excel workbooks with a file location object [page 267]

16.6.1 Associating flat file format with file location object

Associate an existing flat file format with a file location object to use the transfer protocol specified in the file location object to access data.

Note

You can associate an existing file location object with an existing or new flat file format template. The following instructions work with an existing flat file format template.

1. Right-click the name of the applicable flat file format in the object library under *Flat Files* and click *Edit*.
The *File Format Editor <filename>* dialog box opens.
2. Complete the following options in the *Data Files* section:

Option	Description
<i>Location</i>	Name of the file location object.
<i>Delete file after transfer</i>	This option appears only when you choose a file location object for the <i>Location</i> option. ○ Yes: The software deletes the data file located in the local server after it has been used as a source or target in a data flow. ○ No: The software saves the data file located in the local server after it has been used as a source or target in a data flow. You can change this setting in the Source or Target Editor when you set up the file format/file location object pair as a source or target in a data flow.

3. Complete other applicable options related to the file format and then click [Save & Close](#).

Related Information

[Reference Guide: Objects, Description of objects, File format, File format properties and modes](#)

[Source and target objects \[page 277\]](#)

[Manage local and remote data files \[page 257\]](#)

16.6.2 Associating XML, DTD, or JSON schema files

Steps to associate a nested schema with a file location object.

Create a nested schema for DTD, XML, or JSON.

1. Right-click the name of the applicable nested schema template in the object library under [Nested Schemas](#) and click [Edit](#).

The schema opens showing the [General](#) tab, which is view only.

2. Open the [Format](#) tab and complete the following options:

Option	Description
File Location	Name of the file location object.
Delete file after transfer	This option appears only when you choose a file location object for the File Location option. ○ Yes: The software deletes the data file located in the local server after it has been used as a source or target in a data flow. ○ No: The software saves the data file located in the local server after it has been used as a source or target in a data flow. You can change this setting in the source or target editor when you use this object in a data flow.

3. Complete other applicable options related to the file format and then click [Save & Close](#).

When you include the nested schema that you just created in a data flow, the file location object appears in the Source tab.

Related Information

[Reference Guide, Objects, Description of objects, File location objects, Common options](#)

[Manage local and remote data files \[page 257\]](#)

16.6.3 Associating COBOL copybooks with a file location object

Steps to associate a COBOL copybook with a file location object.

i Note

A COBOL copybook object can only be used as a source in a data flow.

1. Right-click the name of the applicable COBOL copybook in the object library under *Nested Schemas* and click *Edit*.

The *Edit COBOL Copybook* dialog opens.

2. Complete the following options in the *Data File* tab:

Option	Description
<i>File Location</i>	Name of the file location object.
<i>Delete file after transfer</i>	This option appears only when you choose a file location object for the <i>File Location</i> option. ○ Yes: The software deletes the data file located in the local server after it has been used as a source in a data flow. ○ No: The software saves the data file located in the local server after it has been used as a source in a data flow. You can change this setting in the source or target editor when you use this object in a data flow. Not applicable for target objects.

3. Complete the remaining options as applicable. Option descriptions are in the *Reference Guide*.

The file location object is associated with the Cobol Copybook.

Related Information

[Reference Guide, Objects, Description of objects, File location objects, Common options](#)

[Creating a new COBOL copybook file format \[page 247\]](#)

[Manage local and remote data files \[page 257\]](#)

16.6.4 Associating Excel workbooks with a file location object

Steps to associate an Excel workbook with a file location object.

i Note

Excel workbooks can only be used as a source in a data flow.

1. Right-click an existing Excel workbook under *Excel Workbooks* in the *Formats* tab of the object library and click *Edit*.

The *Import Excel Workbook* dialog opens.

2. Complete the following options in the *Format* tab:

Option	Description
<i>File Location</i>	Name of the file location object.
<i>Delete file after transfer</i>	This option appears only when you choose a file location object for the <i>File Location</i> option. ◦ Yes: The software deletes the data file located in the local server after it has been used as a source in a data flow. ◦ No: The software saves the data file located in the local server after it has been used as a source in a data flow. You can change this setting in the source or target editor when you use this object in a data flow. Not applicable for target objects.

3. Complete the remaining options as applicable. Option descriptions are in the *Reference Guide*.

The file location object is associated with the Excel workbook.

Related Information

[Reference Guide, Objects, Description of objects, File location objects, Common options](#)
[Creating a Microsoft Excel workbook file format on UNIX \[page 250\]](#)

16.7 Adding a file location object to data flow

Steps to add a file location object to a data flow as a source or target.

You can perform these steps with a format object that does or does not have an associated file location object. If the format object does not have an associated file location object, or you want to edit the associated file location object, you can add or edit the association while performing the steps below.

1. Drag the applicable format object into a data flow on your workspace.
2. Select *Make Source* or *Make Target* from the popup dialog as applicable.
3. Click the name of the object that you just added. If it is a source, the *Source File Editor* opens. If it is a target, the *Target File Editor* opens.

The options that appear vary based on the type of format object you chose (flat file, JSON, XML, for example).

4. In the *Source File Editor* or *Target File Editor*, select an existing file location object from the *Location* drop menu. You can also do this to change the existing association to a different file location object if applicable.
5. Close the *Source File Editor* or *Target File Editor* window.
6. Continue with the data flow setup as applicable.

Related Information

[Creating and defining data flows \[page 275\]](#)

[File location objects \[page 256\]](#)

[Associate file location objects to file formats \[page 263\]](#)

[Manage local and remote data files \[page 257\]](#)

16.8 Using scripts to move files from or to a remote server

Examples of using file location object built-in functions in scripts in a work flow.

Use the two built-in functions, `copy_from_remote_server` and `copy_to_remote_server` in data flow scripts to identify a file location object and a format object instead of using a format with a file location object association. The scripts identify the file location object that contains your file transfer protocol settings, so the software uses your protocol for copying files from and to the remote server. You can also enter a file name using wildcard characters (* and ?) to move a group of files that match the wildcard criteria.

Example

Script file examples

Function	Result
≡ Sample Code <pre>copy_from_remote_system('FTP_Datastore', '*.txt');</pre>	Download files that match the wildcard criteria in the file name from remote server.
≡ Sample Code <pre>copy_from_remote_system('FTP_Datastore', 'mydata.txt');</pre>	Download individual file from remote server.
≡ Sample Code <pre>copy_to_remote_system('FTP_Datastore', 'myremotedata.txt');</pre>	Upload individual file to remote server.
≡ Sample Code <pre>copy_to_remote_system('FTP_Datastore', '*.txt');</pre>	Upload files that match the wildcard criteria in the file name to the remote server.

Related Information

[Reference Guide: Functions and procedures, Descriptions of built-in functions](#)

16.8.1 Moving files to and from Azure containers

Use scripts along with a file location object to move files (called blobs when in a container) from an Azure container to your local directory or to move blobs processed in SAP Data Services into your Azure container.

Use an existing container or create one if it does not exist. The files can be any type. Data Services does not internally manipulate files. Currently, Data Services supports the block blob in the container storage type.

Use a file format to describe a blob file and use it within a data flow to perform extra operations on the file. The file format can also be used in a script to automate upload and to delete the local file.

The following are the high-level steps for uploading files to a container storage blob in Microsoft Azure.

1. Create a storage account in Azure and take note of the primary shared key. For more information, see Microsoft documentation or Microsoft technical support.
2. Create a file location object with the Azure Cloud Storage protocol.

For information about file location objects, see the *Designer Guide*.

3. Create a job in Data Services Designer.
4. Add a script containing the appropriate function to the job.

To move files between remote and local directories, use the following scripts:

- `copy_to_remote_system`
- `copy_from_remote_system`

To access a subfolder in your Azure container, specify the subfolder in the script.

❖ Example

```
copy_to_remote_system('New_FileLocation', '*', '<container_name>/  
<remote_directory>/<sub_folder>')
```

A script that contains this function copies all of the files from the local directory specified in the file location object to the container specified in the same object. When you include the remote directory and subfolder in the script, the function copies all of the files from the local directory to the subfolder specified in the script.

For details about the `copy_to_remote_system` and `copy_from_remote_system` functions, see the descriptions of built-in functions in the *Reference Guide*.

5. Save and run the job.

Related Information

[Azure Cloud Storage protocol](#)

17 Data flows

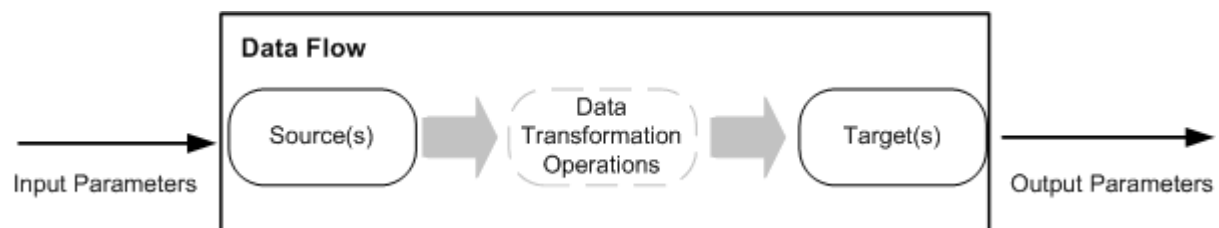
Describes the fundamentals of data flows including data flow objects, using lookups, data flow execution, and auditing.

17.1 What is a data flow?

Data flows extract, transform, and load data.

Everything having to do with data, including reading sources, transforming data, and loading targets, occurs inside a data flow. The lines connecting objects in a data flow represent the flow of data through data transformation steps.

After you define a data flow, you can add it to a job or work flow. From inside a work flow, a data flow can send and receive information to and from other objects through input and output parameters.



→ Remember

Be aware that the data you provide gets placed into trace logs, sample reports, and repositories (side-effect data), and so on. In other words, your data will find its way into places other than output files.

17.1.1 Naming data flows

Data flow names can include alphanumeric characters and underscores (_). They cannot contain blank spaces.

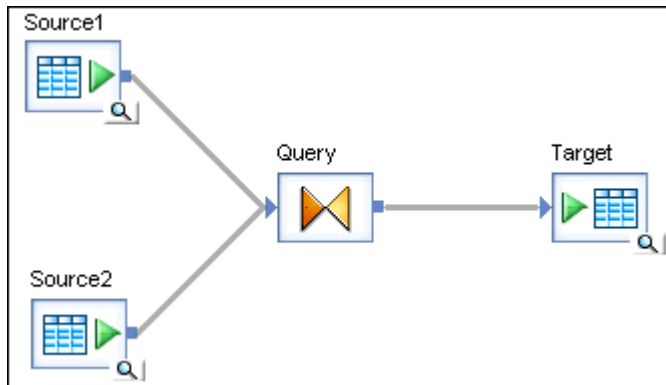
17.1.2 Data flow example

Suppose you want to populate the fact table in your data warehouse with new data from two tables in your source transaction database.

Your data flow consists of the following:

- Two source tables
- A join between these tables, defined in a query transform
- A target table where the new rows are placed

You indicate the flow of data through these components by connecting them in the order that data moves through them. The resulting data flow looks like the following:



17.1.3 Steps in a data flow

Each icon you place in the data flow diagram becomes a step in the data flow.

You can use the following objects as steps in a data flow:

- source
- target
- transforms

The connections you make between the icons determine the order in which the software completes the steps.

Related Information

[Source and target objects \[page 277\]](#)

[Transforms \[page 299\]](#)

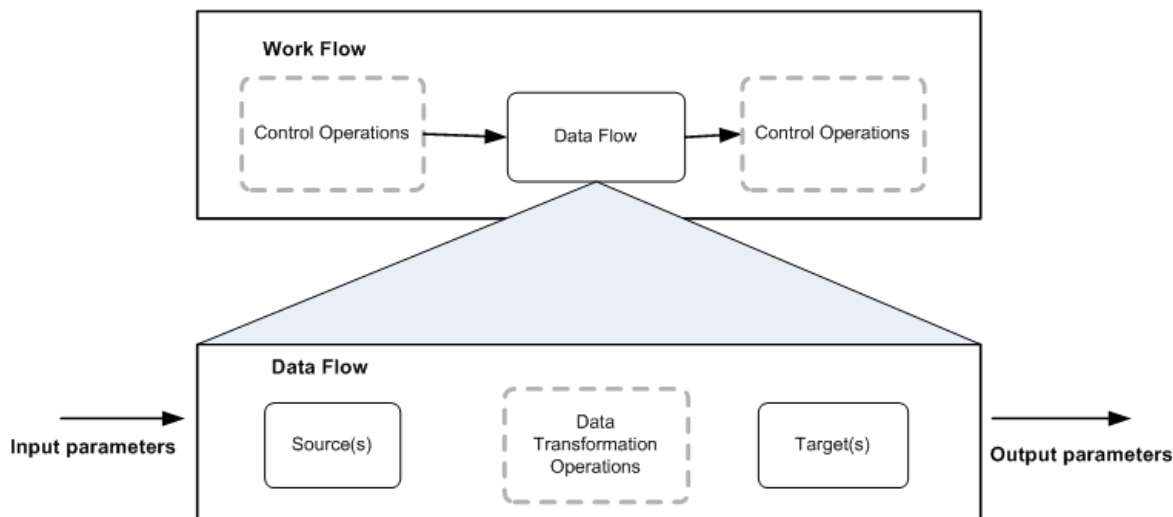
17.1.4 Data flows as steps in work flows

Data flows are closed operations, even when they are steps in a work flow.

Data sets created within a data flow are not available to other steps in the work flow.

A work flow does not operate on data sets and cannot provide more data to a data flow; however, a work flow can do the following:

- Call data flows to perform data movement operations
- Define the conditions appropriate to run data flows
- Pass parameters to and from data flows



17.1.5 Intermediate data sets in a data flow

Each step in a data flow—up to the target definition—produces an intermediate result (for example, the results of a SQL statement containing a WHERE clause), which flows to the next step in the data flow.

The intermediate result consists of a set of rows from the previous operation and the schema in which the rows are arranged. This result is called a data set. This data set may, in turn, be further "filtered" and directed into yet another data set.

17.1.6 Operation codes

Each row in a data set is flagged with an operation code that identifies the status of the row.

The operation codes are as follows:

Operation code	Description
NORMAL	Creates a new row in the target.
	All rows in a data set are flagged as NORMAL when they are extracted from a source. If a row is flagged as NORMAL when loaded into a target, it is inserted as a new row in the target.

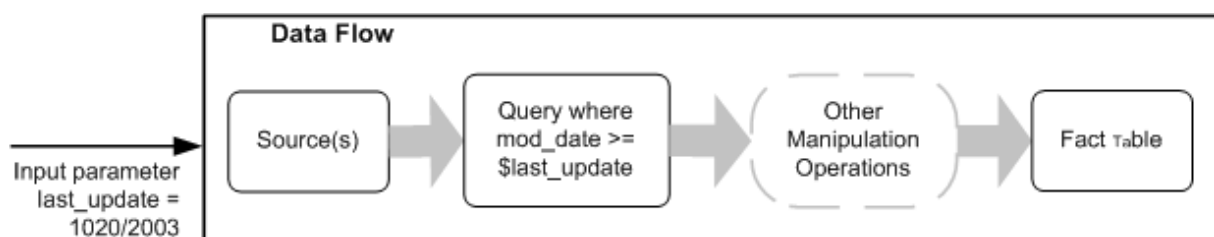
Operation code	Description
INSERT	Creates a new row in the target. Rows can be flagged as INSERT by transforms in the data flow to indicate that a change occurred in a data set as compared with an earlier image of the same data set. The change is recorded in the target separately from the existing data.
DELETE	Is ignored by the target. Rows flagged as DELETE are not loaded. Rows can be flagged as DELETE only by the Map_Operation transform.
UPDATE	Overwrites an existing row in the target. Rows can be flagged as UPDATE by transforms in the data flow to indicate that a change occurred in a data set as compared with an earlier image of the same data set. The change is recorded in the target in the same row as the existing data.

17.1.7 Passing parameters to data flows

Data does not flow outside a data flow, not even when you add a data flow to a work flow. You can, however, pass parameters into and out of a data flow.

Parameters evaluate single values rather than sets of values. When a data flow receives parameters, the steps inside the data flow can reference those parameters as variables.

Parameters make data flow definitions more flexible. For example, a parameter can indicate the last time a fact table was updated. You can use this value in a data flow to extract only rows modified since the last update. The following figure shows the parameter `last_update` used in a query to determine the data set used to load the fact table.



Related Information

[Variables and Parameters \[page 701\]](#)

17.2 Creating and defining data flows

You can create data flows using objects from the object library and the tool palette.

After creating a data flow, you can change its properties.

Related Information

[Changing properties of a data flow \[page 275\]](#)

17.2.1 Defining a new data flow using the object library

1. In the object library, go to the [Data Flows](#) tab.
2. Select the data flow category, right-click and select [New](#).
3. Select the new data flow.
4. Drag the data flow into the workspace for a job or a work flow.
5. Add the sources, transforms, and targets you need.

17.2.2 Defining a new data flow using the tool palette

1. Select the data flow icon in the tool palette.
2. Click the workspace for a job or work flow to place the data flow.

You can add data flows to batch and real-time jobs. When you drag a data flow icon into a job, you are telling the software to validate these objects according the requirements of the job type (either batch or real-time).

3. Add the sources, transforms, and targets you need.

17.2.3 Changing properties of a data flow

After creating a data flow, you can change its properties.

1. Right-click the data flow and select [Properties](#).

The [Properties](#) dialog box opens for the data flow with the [General](#) tab opened.

2. Change desired properties of a data flow.
3. Click [OK](#).

The following table contains descriptions for the options on the General tab of the Properties dialog box.

Option	Description
<i>Execute only once</i>	Specifies that Data Services execute the data flow once in a batch job, and not re-execute the data flow after it completes successfully. There is an exception. When the data flow is contained in a work flow that is a recovery unit, and the recovery unit re-executes but does not complete successfully elsewhere, outside of the recover unit. Because of this exception, we recommend that you do not enable <i>Execute only once</i> for a data flow when a parent work flow is a recovery unit.
<i>Use database links</i>	Specifies whether the data flow uses database links, which are communication paths between one database server and another. Database links allow local users to access data on a remote database, which can be on the local or a remote computer of the same or different database type.
<i>Degree of parallelism</i>	Specifies whether Data Services splits the transforms in a data flow into separate threads to run in parallel, and the maximum number of threads allowed. The default setting is 0, with essentially disables DOP on the local, data flow level. However, a global setting for the Job Server overrides the default of 0. When you set this option to greater than 0, the local setting overrides the global setting. Read about the global setting in Job Server Environment option [page 44].
<i>Cache type</i>	Specify the type of caching Data Services uses to improve performance of operations such as joins, groups, sorts, filtering, lookups, and table comparisons. ◦ <i>In Memory</i>: Choose this value if your data flow processes a small amount of data that can fit in the available memory. ◦ <i>Pageable</i>: This value is the default.

Option	Description
Bypass	Specifies whether Data Services bypasses the execution of a data flow during design time. This option is available at the data flow call level only, for example, when the data flow is in the Designer workspace. ! Restriction The Bypass dropdown list contains only substitution parameters. Therefore, create Bypass substitution parameters before you set this option. For example, in the Substitution Parameter Editor window you might create <code>\$\$BYPASSEnable</code> with a value of Yes and <code>\$\$BYPASSDisable</code> with a value of No. After you finish designing your job, disable bypassing before moving it to production mode. For more details about the bypass option, see Bypassing specific work flows and data flows [page 852].

Related Information

[Creating a new substitution parameter \[page 724\]](#)

[Linked datastores \[page 186\]](#)

[Degree of parallelism](#)

[Data flow](#)

[Cache data](#)

17.3 Source and target objects

A data flow directly reads and loads data using source and target objects.

Source objects define the sources from which you read data.

Target objects define targets to which you write (or load) data.

Related Information

[Source objects \[page 278\]](#)

[Target objects \[page 278\]](#)

17.3.1 Source objects

Source objects contain the data to be read by the data flow for processing.

Source object	Description	Software access
Table	A file formatted with columns and rows as used in relational databases.	Direct or through adapter
Template table	A template table that has been created and saved in another data flow (used in development).	Direct
File	A delimited or fixed-width flat file.	Direct
Document	A file with an application- specific format (not readable by SQL or XML parser).	Through adapter
JSON file	A file formatted with JSON data.	Direct
JSON message	Used as a source in real-time jobs.	Direct
XML file	A file formatted with XML tags.	Direct
XML message	Used as a source in real-time jobs.	Direct

You can also use IDoc messages as real-time sources for SAP applications.

Related Information

[Real-time source and target objects \[page 741\]](#)

[Supplement for SAP: IDoc sources in real-time jobs](#)

17.3.2 Target objects

Target objects receive the processed data that is loaded from the data flow.

Target object	Description	Software access
Document	A file with an application-specific format (not readable by SQL or XML parser).	Through adapter
File	A delimited or fixed-width flat file.	Direct
JSON file	A file formatted in the JSON format.	Direct
JSON message	See Real-time source and target objects [page 741] .	
Nested Template file	A JSON or XML file whose format is based on the preceding transform output (used in development, primarily for debugging data flows).	Direct
Outbound message	See Real-time source and target objects [page 741] .	

Target object	Description	Software access
Table	A file formatted with columns and rows as used in relational databases.	Direct or through adapter
Template table	A table whose format is based on the output of the preceding transform (used in development).	Direct
XML file	A file formatted with XML tags.	Direct
XML message	See Real-time source and target objects [page 741] .	

You can also use IDoc messages as real-time sources for SAP applications.

Related Information

[Supplement for SAP: IDoc targets in real-time jobs](#)

17.3.2.1 Load Triggers tab in target editor

A load trigger operation is a template SQL statement that has placeholders for column and variable values.

Specify load trigger commands in the [Load Triggers](#) tab of the target editor in a data flow.

A load trigger specifies a SQL command that is performed by the database on an INSERT, UPDATE, or DELETE operation. Set the special operation in the load trigger to occur before, after, or instead of normal operations.

The software sets the placeholders at execution time based on the fields in the transform input schema. For each row, the software fills out the template and applies the operation against the target.

i Note

The software does not support load trigger options for Microsoft APS or Azure DW.

Use load triggers in situations such as when you archive updates to a data warehouse or archive incremental updates of aggregated values.

The software does not parse load triggers. Thus, when you specify a load trigger, the software does not parameterize SQL statements. As a result, load times might be higher when you use load triggers.

The software does not validate load triggers.

i Note

If you use an override, you cannot specify auto correct load.

❖ Example

For example, instead of applying an insert of a new sales order row, you use a load trigger that applies inserts and updates of aggregated values of `<sales_per_customer>` and `<sales_per_region>`.

The templates give you a row with `<customer_id>`, `<order_amount>`, `<region_id>`, and so on.

The INSERT and UPDATE statements are:

```
INSERT into order_fact
values ([customer_id], [order_amount]);
UPDATE region_fact
SET order_amount =
order_amount + [order_amount]
WHERE region_id = [region_id];
```

Enter your load triggers manually in the text box in the [Load Triggers](#) tab or drag column names from the [Schema In](#) pane. Enclose column names in curly braces or square brackets. For example, {SalesOffice} or [SalesOffice].

With curly braces, the software encloses the value in quotation marks, if needed. The software does not provide quotes around field names in square brackets. To avoid unintended results, use curly braces for varchar or char column names.

If you insert column names into the SQL statement by dragging the column names, the software inserts square brackets for you. If you require curly braces, manually change the square brackets to curly braces.

The default operations that you select from the [On operation](#) dropdown list are [#insert], [#update], and [#delete].

❖ Example

To delimit a SQL statement, use [#new]. For example:

```
[#insert] [#new]
insert into foo values ([col1], {col2}, ...)
```

Other guidelines for building the SQL statement include the following:

- To specify "before" images, add the suffix .before to the column name. To specify "after" images, add the suffix .after to the column name.
 - The default suffix for UPDATE and INSERT operations is .after.
 - The default suffix for DELETE operations is .before.
- For UPDATE operations, specify both the "before" and the "after" image values. You can specify both images for INSERT and DELETE operations, also, but it is not required.
- Include variables in the SQL statements, but not expressions.
- You can map a batch of SQL statements. Each SQL statement is separated by a new separator ([#new]).

The following statement is an example for mapping insert SQL:

```
INSERT into log_table values ({col1}, {col2})
[#new]
[#insert] [#new]
delete from alt_junk where . . .
```

17.3.2.2 Setting DB2 10.5 for column based target table

When you select *Column store* for the DB2 option *Table Type*, also complete specific prerequisite tasks.

A specific target set up option is *Table Type* where you can select either *Row store* or *Column store*. This task takes you through the prerequisite steps for DB2 when you select *Column store*. The software supports column organized tables only on Linux and AIX:

- Linux: x86-x64, Intel, and AMD processors
- AIX: POWER processors

There are two ways to enable column store in Linux and AIX:

- Default table organization: Set the registry variable DB2_WORKLOAD to ANALYTICS.
- Configure database for analytics workload: If your DB2_WORKLOAD cannot be set to ANALYTICS, create and optimally configure the database for analytics workload by following these steps.

1. Set the default table organization setting for user tables `dft_table_org` to COLUMN.

Set `dft_table_org` to column so that new tables are organized by column by default. If you do not set `dft_table_org` to column, you have to set the ORGANIZE BY COLUMN clause for each CREATE TABLE statement.

2. Set the default degree database configuration parameter `dft_degree` to ANY.
3. Set the default extent size database configuration parameter `dft_extent_sz` to 4.
4. Ensure that the database configuration parameters for sort heap `sortheap` and sort heap threshold for shared sorts `sheapthres_shr` are **not** set to *AUTOMATIC*.

Consider increasing `sortheap` and `sheapthres_shr` values significantly for analytics workloads. A reasonable starting point is to set `sheapthres_shr` to the size of the buffer pool across all buffer pools. Set `sortheap` to a fraction, for example 1/20, of `sheapthres_shr` to enable concurrent sort operations.

5. Set the utility heap size database configuration parameter `util_heap_sz` to 1,000,000 pages and *AUTOMATIC* to address the resource needs of the LOAD command.
 - If the database server has at least 128 GB of memory, set `util_heap_sz` to 4,000,000 pages.
 - If concurrent load operations are running, increase the value of `util_heap_sz` to accommodate higher memory requirements.
6. Set the automatic reorganization database configuration parameter `auto_reorg` to ON.
7. Ensure that the database manager configuration parameter `sheapthres` to ON.

Note

This setting applies to all databases in the instance.

8. Ensure that intraquery parallelism, which is required to access column-organized tables, is enabled.

Enable intraquery parallelism at the instance level or the database level. For details, see Intraquery parallelism and inrapartition parallelism in your DB2 documentation.

9. Enable concurrency control on the SYSDEFAULTMANAGEDSUBCLASS service subclass by issuing the following statement: `ALTER THRESHOLD SYSDEFAULTCONCURRENT ENABLE`.

Additional restrictions

Schemas that include column-organized tables cannot be transported.

For data replication with column-organized tables as either source or target, there are some sql statements that do not support column-organized tables as source and target tables:

- SET INTEGRITY
- CREATE TRIGGER
- CREATE EVENT MONITOR
- CREATE INDEX AND ALTER INDEX

17.3.3 Adding source or target objects to data flows

Fulfill the following prerequisites before using a source or target object in a data flow:

For	Prerequisite
Tables accessed directly from a database	Define a database datastore and import table metadata.
Template tables	Define a database datastore.
Files	Define a file format and import the file.
XML files and messages	Import an XML file format.
Objects accessed through an adapter	Define an adapter datastore and import object metadata.

1. Open the data flow in which you want to place the object.
2. If the object library is not already open, select **Tools > Object Library** to open it.
3. Select the appropriate object library tab: Choose the Formats tab for flat files, DTDs, JSNs, or XML Schemas, or choose the Datastores tab for database and adapter objects.
4. Select the object you want to add as a source or target. (Expand collapsed lists by clicking the plus sign next to a container icon.)

For a new template table, select the Template Table icon from the tool palette.

For a new JSON or XML template file, select the Nested Schemas Template icon from the tool palette.

5. Drop the object in the workspace.
6. For objects that can be either sources or targets, when you release the cursor, a popup menu appears. Select the kind of object to make.

For new template tables and XML template files, when you release the cursor, a secondary window appears. Enter the requested information for the new template object. Names can include alphanumeric characters and underscores (_). Template tables cannot have the same name as an existing table within a datastore.

7. The source or target object appears in the workspace.
8. Click the object name in the workspace

The software opens the editor for the object. Set the options you require for the object.

i Note

Ensure that any files that reference flat file, DTD, JSON, or XML Schema formats are available from the Job Server where the job will be run and specify the file location relative to this computer.

Related Information

[Database datastores \[page 93\]](#)

[Flat file formats \[page 221\]](#)

[Importing a DTD or XML Schema format \[page 671\]](#)

[Adding a file location object to data flow \[page 268\]](#)

17.3.4 Template tables

During the initial design of an application, you might find it convenient to use template tables to represent database tables.

With template tables, you do not have to initially create a new table in your DBMS and import the metadata into the software. Instead, the software automatically creates the table in the database with the schema defined by the data flow when you execute a job.

After creating a template table as a target in one data flow, you can use it as a source in other data flows. Though a template table can be used as a source table in multiple data flows, it can only be used as a target in one data flow.

Template tables are particularly useful in early application development when you are designing and testing a project. If you modify and save the data transformation operation in the data flow where the template table is a target, the schema of the template table automatically changes. Any updates to the schema are automatically made to any other instances of the template table. During the validation process, the software warns you of any errors such as those resulting from changing the schema.

17.3.4.1 Creating a target template table

1. Use one of the following methods to open the [Create Template](#) window:
 - From the tool palette:
 - Click the template table icon. 
 - Click inside a data flow to place the template table in the workspace.
 - In the [Create Template](#) window, select a datastore.
 - From the object library:
 - Expand a datastore.
 - Click the template table icon and drag it to the workspace.

2. In the [Create Template](#) window, enter a table name.

If you are using Netezza 7.x, you may enter a schema name in the schema box to limit the template tables to a particular schema. If you leave the schema name blank, the template tables are limited to the default schema.

3. Click [OK](#).

The table appears in the workspace as a template table icon.

4. Connect the template table to the data flow as a target (usually a Query transform).
5. In the Query transform, map the Schema In columns that you want to include in the target table.
6. From the [Project](#) menu select [Save](#).

In the workspace, the template table's icon changes to a target table icon and the table appears in the object library under the datastore's list of tables.

After you are satisfied with the design of your data flow, save it. When the job is executed, software uses the template table to create a new table in the database you specified when you created the template table. Once a template table is created in the database, you can convert the template table in the repository to a regular table.

17.3.5 Converting template tables to regular tables

You must convert template tables to regular tables to take advantage of some features such as bulk loading. Other features, such as exporting an object, are available for template tables.

i Note

Once a template table is converted, you can no longer alter the schema.

17.3.5.1 Converting a template table into a regular table from the object library

1. Open the object library and go to the [Datastores](#) tab.
2. Click the plus sign (+) next to the datastore that contains the template table you want to convert.

A list of objects appears.

3. Click the plus sign (+) next to [Template Tables](#).

The list of template tables appears.

4. Right-click a template table you want to convert and select [Import Table](#).

The software converts the template table in the repository into a regular table by importing it from the database. To update the icon in all data flows, choose [View](#) [Refresh](#). In the datastore object library, the table is now listed under Tables rather than Template Tables.

17.3.5.2 Converting a template table into a regular table from a data flow

1. Open the data flow containing the template table.
2. Right-click on the template table you want to convert and select *Import Table*.

After a template table is converted into a regular table, you can no longer change the table's schema.

17.4 Understanding column propagation

You can use the *Propagate Column From* command in a data flow to add an existing column from an upstream source or transform through intermediate objects to a selected endpoint.

Columns are added in each object with no change to the data type or other attributes. When there is more than one possible path between the starting point and ending point, you can specify the route for the added columns.

Column propagation is a pull-through operation. The *Propagate Column From* command is issued from the object where the column is needed. The column is pulled from the selected upstream source or transform and added to each of the intermediate objects as well as the selected endpoint object.

For example, in the data flow below, the `Employee` source table contains employee name information as well as employee ID, job information, and hire dates. The `Name_Cleanse` transform is used to standardize the employee names. Lastly, the data is output to an XML file called `Employee_Names`.



After viewing the output in the `Employee_Names` table, you realize that the middle initial (`minit` column) should be included in the output. You right-click the top-level schema of the `Employee_Names` table and select *Propagate Column From*. The *Propagate Column to Employee_Names* window appears.

In the left pane of the *Propagate Column to Employee_Names* window, select the `Employee` source table from the list of objects. The list of output columns displayed in the right pane changes to display the columns in the schema of the selected object. Select the `MINIT` column as the column you want to pull through from the source, and then click *Propagate*.

The `minit` column schema is carried through the `Query` and `Name_Cleanse` transforms to the `Employee_Names` table.

Characteristics of propagated columns are as follows:

- The *Propagate Column From* command can be issued from the top-level schema of either a transform or a target.

- Columns are added in each object with no change to the data type or other attributes. Once a column is added to the schema of an object, the column functions in exactly the same way as if it had been created manually.
- The propagated column is added at the end of the schema list in each object.
- The output column name is auto-generated to avoid naming conflicts with existing columns. You can edit the column name, if desired.
- Only columns included in top-level schemas can be propagated. Columns in nested schemas cannot be propagated.
- A column can be propagated more than once. Any existing columns are shown in the right pane of the [Propagate Column to](#) window in the *Already Exists In* field. Each additional column will have a unique name.
- Multiple columns can be selected and propagated in the same operation.

i Note

You cannot propagate a column through a `Hierarchy_Flattening` transform or a `Table_Comparison` transform.

17.4.1 Adding columns within a dataflow

You can add a column from an upstream source or transform, through intermediate objects, to a selected endpoint using the propagate command.

Columns are added in each object with no change to the data type or other attributes.

To add columns within a data flow:

1. In the downstream object where you want to add the column (the endpoint), right-click the top-level schema and click [Propagate Column From](#).
The [Propagate Column From](#) can be issued from the top-level schema in a transform or target object.
2. In the left pane of the [Propagate Column to](#) window, select the upstream object that contains the column you want to map.
The available columns in that object are displayed in the right pane along with a list of any existing mappings from that column.
3. In the right pane, select the column you wish to add and click either [Propagate](#) or [Propagate and Close](#). One of the following occurs:
 - If there is a single possible route, the selected column is added through the intermediate transforms to the downstream object.
 - If there is more than one possible path through intermediate objects, the [Choose Route to](#) dialog displays. This may occur when your data flow contains a Query transform with multiple input objects. Select the path you prefer and click [OK](#).

17.4.2 Propagating columns in a data flow containing a Merge transform

Propagation sends column changes downstream.

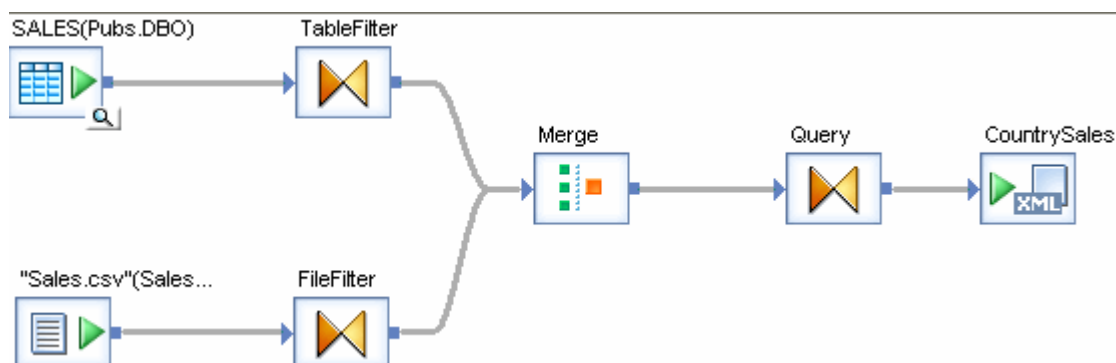
Invalid data flows that contain two or more sources which are merged using a Merge transform, the schema of the inputs into the Merge transform must be identical. All sources must have the same schema, including:

- the same number of columns
- the same column names
- like columns must have the same data type

In order to maintain a valid data flow when propagating a column through a Merge transform, you must make sure to meet this restriction.

When you propagate a column and a Merge transform falls between the starting point and ending point, a message warns you that after the propagate operation completes the data flow will be invalid because the input schemas in the Merge transform will not be identical. If you choose to continue with the column propagation operation, you must later add columns to the input schemas in the Merge transform so that the data flow is valid.

For example, in the data flow shown below, the data from each source table is filtered and then the results are merged in the Merge transform.



If you choose to propagate a column from the `SALES (Pubs.DBO)` source to the `CountrySales` target, the column would be added to the `TableFilter` schema but not to the `FileFilter` schema, resulting in differing input schemas in the Merge transform and an invalid data flow.

In order to maintain a valid data flow, when propagating a column through a Merge transform you may want to follow a multi-step process:

1. Ensure that the column you want to propagate is available in the schemas of all the objects that lead into the Merge transform on the upstream side. This ensures that all inputs to the Merge transform are identical and the data flow is valid.
2. Propagate the column on the downstream side of the Merge transform to the desired endpoint.

17.5 Lookup tables and the lookup_ext function

Lookup tables contain data that other tables reference.

Typically, lookup tables can have the following kinds of columns:

- **Lookup column**—Use to match a row(s) based on the input values. You apply operators such as =, >, <, ~ to identify a match in a row. A lookup table can contain more than one lookup column.
- **Output column**—The column returned from the row that matches the lookup condition defined for the lookup column. A lookup table can contain more than one output column.
- **Return policy column**—Use to specify the data to return in the case where multiple rows match the lookup condition(s).

Use the lookup_ext function to retrieve data from a lookup table based on user-defined lookup conditions that match input data to the lookup table data. Not only can the lookup_ext function retrieve a value in a table or file based on the values in a different source table or file, but it also provides extended functionality that lets you do the following:

- Return multiple columns from a single lookup
- Choose from more operators, including pattern matching, to specify a lookup condition
- Specify a return policy for your lookup
- Call lookup_ext in scripts and custom functions (which also lets you reuse the lookup(s) packaged inside scripts)
- Define custom SQL using the SQL_override parameter to populate the lookup cache, which is useful for narrowing large quantities of data to only the sections relevant for your lookup(s)
- Call lookup_ext using the function wizard in the query output mapping to return multiple columns in a Query transform
- Choose a caching strategy, for example decide to cache the whole lookup table in memory or dynamically generate SQL for each input record
- Use lookup_ext with memory datastore tables or persistent cache tables. The benefits of using persistent cache over memory tables for lookup tables are:
 - Multiple data flows can use the same lookup table that exists on persistent cache.
 - The software does not need to construct the lookup table each time a data flow uses it.
 - Persistent cache has no memory constraints because it is stored on disk and the software quickly pages it into memory.
- Use pageable cache (which is not available for the lookup and lookup_seq functions)
- Use expressions in lookup tables and return the resulting values

For a description of the related functions lookup and lookup_seq, see the *Reference Guide*.

Related Information

[Reference Guide: Functions and Procedures, lookup_ext](#)

[Performance Optimization Guide: Using Caches, Caching data](#)

17.5.1 Accessing the lookup_ext editor

Lookup_ext has its own graphic editor.

There are two ways to invoke the editor:

- Add a new function call inside a Query transform—Use this option if you want the lookup table to return more than one column.
- From the Mapping tab in a query or script function.

17.5.1.1 Adding a new function call

1. In the Query transform *Schema out* pane, without selecting a specific output column right-click in the pane and select *New Function Call*.
2. Select the Function category *Lookup Functions* and the Function name *lookup_ext*.
3. Click *Next* to invoke the editor.

In the Output section, you can add multiple columns to the output schema.

An advantage of using the new function call is that after you close the lookup_ext function window, you can reopen the graphical editor to make modifications (right-click the function name in the schema and select *Modify Function Call*).

17.5.1.2 Invoking the lookup_ext editor from the Mapping tab

1. Select the output column name.
2. On the *Mapping* tab, click *Functions*.
3. Select the *Function category Lookup Functions* and the *Function name lookup_ext*.
4. Click *Next* to invoke the editor.

In the Output section, *Variable* replaces *Output column name*. You can define one output column that will populate the selected column in the output schema. When lookup_ext returns more than one output column, use variables to store the output values, or use lookup_ext as a new function call as previously described in this section.

With functions used in mappings, the graphical editor isn't available, but you can edit the text on the *Mapping* tab manually.

17.5.2 Example: Defining a simple lookup_ext function

This procedure describes the process for defining a simple lookup_ext function using a new function call. The associated example illustrates how to use a lookup table to retrieve department names for employees.

For details on all the available options for the lookup_ext function, see the *Reference Guide*.

1. In a data flow, open the Query editor.
2. From the *Schema in* pane, drag the ID column to the *Schema out* pane.
3. Select the ID column in the *Schema out* pane, right-click, and click *New Function Call*. Click *Insert Below*.
4. Select the *Function category Lookup Functions* and the *Function name lookup_ext* and click *Next*.
The lookup_ext editor opens.
5. In the *Lookup_ext - Select Parameters* window, select a lookup table:
 - a. Next to the *Lookup table* text box, click the drop-down arrow and double-click the datastore, file format, or current schema that includes the table.
 - b. Select the lookup table and click *OK*.In the example, the lookup table is a file format called ID_lookup.txt that is in D:\Data.
6. For the *Cache spec*, the default of PRE_LOAD_CACHE is useful when the number of rows in the table is small or you expect to access a high percentage of the table values.
NO_CACHE reads values from the lookup table for every row without caching values. Select DEMAND_LOAD_CACHE when the number of rows in the table is large and you expect to frequently access a low percentage of table values or when you use the table in multiple lookups and the compare conditions are highly selective, resulting in a small subset of data.
7. To provide more resources to execute the lookup_ext function, select *Run as a separate process*. This option creates a separate child data flow process for the lookup_ext function when the software executes the data flow.
8. Define one or more conditions. For each, add a lookup table column name (select from the drop-down list or drag from the *Parameter* pane), select the appropriate operator, and enter an expression by typing, dragging, pasting, or using the Smart Editor (click the icon in the right column).
In the example, the condition is `ID_DEPT = Employees.ID_DEPT`.
9. Define the output. For each output column:
 - a. Add a lookup table column name.
 - b. Optionally change the default value from NULL.
 - c. Specify the *Output column name* by typing, dragging, pasting, or using the Smart Editor (click the icon in the right column).In the example, the output column is ID_DEPT_NAME.
10. If multiple matches are possible, specify the ordering and set a return policy (default is *MAX*) to select one match. To order the output, enter the column name(s) in the *Order by* list.

Example

The following example illustrates how to use the lookup table ID_lookup.txt to retrieve department names for employees.

The Employees table is as follows:

ID	NAME	ID_DEPT
SSN111111111	Employee1	10
SSN222222222	Employee2	10

ID	NAME	ID_DEPT
TAXID33333333	Employee3	20

The lookup table ID_lookup.txt is as follows:

ID_DEPT	ID_PATTERN	ID_RETURN	ID_DEPT_NAME
10	ms(SSN*)	=substr(ID_Pattern,4,20)	Payroll
20	ms(TAXID*)	=substr(ID_Pattern,6,30)	Accounting

The lookup_ext editor would be configured as follows.

Lookup_ext - Select Parameters

Lookup table: Cache spec: ☐ Run as a separate process

Available parameters:

- Parameter
 - Lookup Table
 - "D:\Data\ID_lookup.txt"(Lookup
 - ID_DEPT
 - ID_PATTERN
 - ID_RETURN
 - ID_DEPT_NAME
 - Input Schema
 - Employees
 - ID
 - NAME
 - ID_DEPT
 - Variables

Condition:

Column in Lookup table	Op.(%)	Expression
ID_DEPT	=	Employees.ID_DEPT
*		

Output:

Column in Lookup table	Expression?	Default	Output column name
ID_DEPT_NAME	☒	NULL	ID_DEPT_NAME
*	☐	NULL	

Order by:

Column in Lookup table
*

Return policy:

(Optional) If multiple rows satisfy the lookup conditions, they are ordered by the list of columns here, and the row that satisfies the "Return policy" is returned.

Custom SQL

< Back Finish Cancel

Related Information

[Example: Defining a complex lookup_ext function \[page 292\]](#)

17.5.3 Example: Defining a complex lookup_ext function

This procedure describes the process for defining a complex lookup_ext function using a new function call. The associated example uses the same lookup and input tables as in the [Example: Defining a simple lookup_ext function \[page 289\]](#) This example illustrates how to extract and normalize employee ID numbers.

For details on all the available options for the lookup_ext function, see the *Reference Guide*.

1. In a data flow, open the Query editor.
2. From the *Schema in* pane, drag the ID column to the *Schema out* pane. Do the same for the Name column.
3. In the *Schema out* pane, right-click the Name column and click *New Function Call*. Click *Insert Below*.
4. Select the *Function category Lookup Functions* and the *Function name lookup_ext* and click *Next*.
5. In the *Lookup_ext - Select Parameters* window, select a lookup table:
In the example, the lookup table is in the file format ID_lookup.txt that is in D:\Data.
6. Define one or more conditions.
In the example, the condition is `ID_PATTERN ~ Employees.ID`.
7. Define the output. For each output column:
 - a. Add a lookup table column name.
 - b. If you want the software to interpret the column in the lookup table as an expression and return the calculated value, select the *Expression* check box.
 - c. Optionally change the default value from NULL.
 - d. Specify the *Output column name(s)* by typing, dragging, pasting, or using the Smart Editor (click the icon in the right column).

In the example, the output columns are ID_RETURN and ID_DEPT_NAME.

Example

Extract and normalize employee ID numbers

In this example, you want to extract and normalize employee Social Security numbers and tax identification numbers that have different prefixes. You want to remove the prefixes, thereby normalizing the numbers. You also want to identify the department from where the number came. The data flow has one source table Employees, a query configured with lookup_ext, and a target table.

Configure the lookup_ext editor as in the following graphic.

Lookup_ext - Select Parameters

Lookup table: Cache spec: ☐ Run as a separate process

Available parameters:

- Parameter
 - Lookup Table
 - "D:\Data\ID_lookup.txt"(Lookup
 - ID_DEPT
 - ID_PATTERN
 - ID_RETURN
 - ID_DEPT_NAME
 - Input Schema
 - Employees
 - ID
 - NAME
 - ID_DEPT
 - Variables

Condition:

Column in Lookup table	Op.(%)	Expression
ID_PATTERN	~	Employees.ID
*		

Output:

Column in Lookup table	Expression?	Default	Output column name
ID_RETURN	☒	NULL	ID_RETURN
ID_DEPT_NAME	☐	NULL	ID_DEPT_NAME
*	☐	NULL	

Order by:

Column in Lookup table
*

Return policy:

(Optional) If multiple rows satisfy the lookup conditions, they are ordered by the list of columns here, and the row that satisfies the "Return policy" is returned.

Custom SQL

< Back Finish Cancel

The lookup condition is `ID_PATTERN ~ Employees.ID`.

The software reads each row of the source table Employees, then checks the lookup table ID_lookup.txt for all rows that satisfy the lookup condition.

The operator ~ means that the software will apply a pattern comparison to Employees.ID. When it encounters a pattern in ID_lookup.ID_PATTERN that matches Employees.ID, the software applies the expression in ID_lookup.ID_RETURN. In this example, Employee1 and Employee2 both have IDs that match the pattern ms(SSN*) in the lookup table. the software then applies the expression `=substr (ID_PATTERN, 4, 20)` to the data, which extracts from the matched string (Employees.ID) a substring of up to 20 characters starting from the 4th position. The results for Employee1 and Employee2 are 111111111 and 222222222, respectively.

For the output of the ID_RETURN lookup column, the software evaluates ID_RETURN as an expression because the [Expression](#) box is checked. In the lookup table, the column ID_RETURN contains the expression `=substr (ID_PATTERN, 4, 20)`. ID_PATTERN in this expression refers to the lookup table column ID_PATTERN. When the lookup condition `ID_PATTERN ~ Employees.ID` is true, the software evaluates the expression. Here the software substitutes the placeholder ID_PATTERN with the actual Employees.ID value.

The output also includes the ID_DEPT_NAME column, which the software returns as a literal value (because the Expression box is not checked). The resulting target table is as follows:

ID	NAME	ID_RETURN	ID_DEPT_NAME
SSN111111111	Employee1	111111111	Payroll
SSN222222222	Employee2	222222222	Payroll
TAXID333333333	Employee3	333333333	Accounting

Related Information

[Example: Defining a simple lookup_ext function \[page 289\]](#)

[Accessing the lookup_ext editor \[page 289\]](#)

[Reference Guide: Functions and Procedures, lookup_ext](#)

[Reference Guide: Functions and Procedures, match_simple](#)

17.6 Data flow execution

A data flow is a declarative specification from which the software determines the correct data to process.

For example in data flows placed in batch jobs, the transaction order is to extract, transform, then load data into a target. Data flows are similar to SQL statements. The specification declares the desired output.

The software executes a data flow each time the data flow occurs in a job. However, you can specify that a batch job execute a particular data flow only one time. In that case, the software only executes the first occurrence of the data flow; the software skips subsequent occurrences in the job.

You might use this feature when developing complex batch jobs with multiple paths, such as jobs with try/catch blocks or conditionals, and you want to ensure that the software only executes a particular data flow one time.

Related Information

[Creating and defining data flows \[page 275\]](#)

17.6.1 Push down operations to the database server

From the information in the data flow specification, the software produces output while optimizing performance.

For example, for SQL sources and targets, the software creates database-specific SQL statements based on a job's data flow diagrams. To optimize performance, the software pushes down as many transform operations as possible to the source or target database and combines as many operations as possible into one request to the database. For example, the software tries to push down joins and function evaluations. By pushing down

operations to the database, the software reduces the number of rows and operations that the engine must process.

Data flow design influences the number of operations that the software can push to the source or target database. Before running a job, you can examine the SQL that the software generates and alter your design to produce the most efficient results.

You can use the Data_Transfer transform to push down resource-intensive operations anywhere within a data flow to the database. Resource-intensive operations include joins, GROUP BY, ORDER BY, and DISTINCT.

Related Information

[Performance Optimization Guide: Maximizing push-down operations](#)

Reference Guide: Data_Transfer

17.6.2 Distributed data flow execution

The software provides capabilities to distribute CPU-intensive and memory-intensive data processing work (such as join, grouping, table comparison and lookups) across multiple processes and computers.

This work distribution provides the following potential benefits:

- Better memory management by taking advantage of more CPU resources and physical memory
- Better job performance and scalability by using concurrent sub data flow execution to take advantage of grid computing

You can create sub data flows so that the software does not need to process the entire data flow in memory at one time. You can also distribute the sub data flows to different job servers within a server group to use additional memory and CPU resources.

Use the following features to split a data flow into multiple sub data flows:

- [Run as a separate process](#) option on resource-intensive operations that include the following:
 - Hierarchy_Flattening transform
 - Associate transform
 - Country ID transform
 - Global Address Cleanse transform
 - Global Suggestion Lists transform
 - Match Transform
 - United States Regulatory Address Cleanse transform
 - User-Defined transform
 - Query operations that are CPU-intensive and memory-intensive:
 - Join
 - GROUP BY
 - ORDER BY
 - DISTINCT
 - Table_Comparison transform

- Lookup_ext function
- Count_distinct function
- Search_replace function

If you select the Run as a separate process option for multiple operations in a data flow, the software splits the data flow into smaller sub data flows that use separate resources (memory and computer) from each other. When you specify multiple Run as a separate process options, the sub data flow processes run in parallel.

- Data_Transfer transform

With this transform, the software does not need to process the entire data flow on the Job Server computer. Instead, the Data_Transfer transform can push down the processing of a resource-intensive operation to the database server. This transform splits the data flow into two sub data flows and transfers the data to a table in the database server to enable the software to push down the operation.

Related Information

[Performance Optimization Guide: Splitting a data flow into sub data flows](#)

[Performance Optimization Guide: Data_Transfer transform for push-down operations](#)

17.6.3 Load balancing

You can distribute the execution of a job or a part of a job across multiple Job Servers within a Server Group to better balance resource-intensive operations.

You can specify the following values on the *Distribution level* option when you execute a job:

- Job level—A job can execute on an available Job Server.
- Data flow level—Each data flow within a job can execute on an available Job Server.
- Sub data flow level—An resource-intensive operation (such as a sort, table comparison, or table lookup) within a data flow can execute on an available Job Server.

Related Information

[Performance Optimization Guide: Using grid computing to distribute data flows execution](#)

17.6.4 Caches

The software provides the option to cache data in memory to improve operations such as the following in your data flows.

Operation	Description
Joins	Because an inner source of a join must be read for each row of an outer source, you might want to cache a source when it is used as an inner source in a join.
Table comparisons	Because a comparison table must be read for each row of a source, you might want to cache the comparison table.
Lookups	Because a lookup table might exist on a remote database, you might want to cache it in memory to reduce access times.

The software provides the following types of caches that your data flow can use for all of the operations it contains:

Cache	Description
In-memory	Use in-memory cache when your data flow processes a small amount of data that fits in memory.
Pageable cache	Use a pageable cache when your data flow processes a very large amount of data that does not fit in memory.

If you split your data flow into sub data flows that each run on a different Job Server, each sub data flow can use its own cache type.

Related Information

[Cache data](#)

17.7 Audit Data Flow overview

You can audit objects within a data flow to collect run time audit statistics.

You can perform the following tasks with this auditing feature:

- Collect audit statistics about data read into a job, processed by various transforms, and loaded into targets.
- Define rules about the audit statistics to determine if the correct data is processed.
- Generate notification of audit failures.
- Query the audit statistics that persist in the repository.

Related Information

[Using Auditing \[page 335\]](#)

18 Transforms

Transforms operate on data sets by manipulating input sets and producing one or more output sets.

By contrast, functions operate on single values in specific columns in a data set. Many built-in transforms are available from the object library on the [Transforms](#) tab.

The transforms that you can use depend on the software package that you have purchased. (If a transform belongs to a package that you have not purchased, it is disabled and cannot be used in a job.)

The software has the following transform categories:

- **Data Integrator:** Transforms that extract, transform, and load data. These transforms help ensure data integrity and maximize developer productivity for loading and updating data in a warehouse environment.
- **Data Quality:** Transforms that improve the quality of your data. These transforms can parse, standardize, correct, enrich, match, and consolidate your customer and operational information assets.
- **Platform:** Transforms that perform general data movement operations. These transforms generate, map and merge rows from two or more sources, create SQL query operations (expressions, lookups, joins, and filters), perform conditional splitting, and mask personal data to keep sensitive data relevant, anonymous, and secure.
- **Text Data Processing:** Transforms that extract specific information from your text. These transforms parse large volumes of text so you can identify and extract entities and facts such as customers, products, locations, and financial information relevant to your organization.

Transform Category	Transform	Description
Data Integrator	Data Transfer	Allows a data flow to split its processing into two sub data flows and push down resource-consuming operations to the database server.
	Date Generation	Generates a column filled with date values based on the start and end dates and increment that you provide.
	Effective Date	Generates an additional "effective to" column based on the "effective date" of the primary key.
	Hierarchy Flattening	Flattens hierarchical data into relational tables so that it can participate in a star schema. Hierarchy flattening can be both vertical and horizontal.
	History Preserving	Converts rows flagged as UPDATE to UPDATE plus INSERT, so that the original values are preserved in the target. You specify in which column to look for updated data.
	Key Generation	Generates new keys for source data, starting from a value based on existing keys in the table you specify.
	Map CDC Operation	Sorts input data, maps output data, and resolves before- and after-images for UPDATE rows. While commonly used to support Oracle changed-data capture, this transform supports any data stream if its input requirements are met.
	Pivot (Columns to Rows)	Rotates the values in specified columns to rows. (Also see Reverse Pivot.)

Transform Category	Transform	Description
Data Quality	Reverse Pivot (Rows to Columns)	Rotates the values in specified rows to columns.
	Table Comparison	Compares two data sets and produces the difference between them as a data set with rows flagged as INSERT and UPDATE.
	XML Pipeline	Processes large XML inputs in small batches.
	Associate	Combine the results of two or more Match transforms, two or more Associate transforms, or any combination of the two, to find matches across match sets.
	Country ID	Parses input data and then identifies the country of destination for each record.
	Data Cleanse	Identifies and parses name, title, and firm data, phone numbers, Social Security numbers, dates, and e-mail addresses. It can assign gender, add pre-names, generate Match standards, and convert input sources to a standard format. It can also parse and manipulate various forms of international data, as well as operational and product data.
	DSF2 Walk Sequencer	Adds delivery sequence information to your data, which you can use with pre-sorting software to qualify for walk-sequence discounts.
	Geocoder	Uses geographic coordinates, addresses, and point-of-interest (POI) data to append address, latitude and longitude, census, and other information to your records.
	Global Address Cleanse	Identifies, parses, validates, and corrects global address data, such as primary number, primary name, primary type, directional, secondary identifier, and secondary number.
	Global Suggestion Lists	Completes and populates addresses with minimal data, and it can offer suggestions for possible matches.
Platform	Match	Identifies matching records based on your business rules. Also performs candidate selection, unique ID, best record, and other operations.
	USA Regulatory Address Cleanse	Identifies, parses, validates, and corrects USA address data according to the U.S. Coding Accuracy Support System (CASS).
	Case	Simplifies branch logic in data flows by consolidating case or decision making logic in one transform. Paths are defined in an expression table.
	Data Mask	Uses data masking techniques to disguise or hide personal information contained in your databases (for example, bank account numbers, credit card numbers, and income). You can use the following techniques in Data Mask: • Character replacement • Number variance • Date variance • Pattern variance • Number generalization • Date generalization Data masking maintains data relevancy and relationships while keeping client information confidential and anonymous, and helps support your data protection policies.

Transform Category	Transform	Description
	Map Operation	Modifies data based on current operation codes and mapping expressions. The operation codes can then be converted between data manipulation operations.
	Merge	Unifies rows from two or more sources into a single target.
	Query	Retrieves a data set that satisfies conditions that you specify. A query transform is similar to a SQL SELECT statement.
	Row Generation	Generates a column filled with integer values starting at zero and incrementing by one to the end value you specify.
	SQL	Performs the indicated SQL query operation.
	User Defined	Does just about anything that you can write Python code to do. For example, you can use the transform to create new records and data sets or populate a field with a specific value.
	Validation	Ensures that the data at any stage in the data flow meets your criteria. You can filter out or replace data that fails your criteria.
Text Data Processing	Entity Extraction	Extracts information (entities and facts) from any text, HTML, XML , or binary format content such as PDF.

Related Information

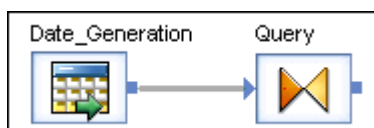
[Reference Guide: Transforms](#)

18.1 Adding a transform to a data flow

You can use the Designer to add transforms to data flows.

1. Open a data flow object.
2. Open the object library if it is not already open and click the [Transforms](#) tab.
3. Select the transform or transform configuration that you want to add to the data flow.
4. Drag the transform or transform configuration icon into the data flow workspace. If you selected a transform that has available transform configurations, a drop-down menu prompts you to select a transform configuration.
5. Draw the data flow connections.

To connect a source to a transform, click the square on the right edge of the source and drag the cursor to the arrow on the left edge of the transform.



Continue connecting inputs and outputs as required for the transform.

- The input for the transform might be the output from another transform or the output from a source; or, the transform may not require source data.
- You can connect the output of the transform to the input of another transform or target.

6. Double-click the name of the transform.

This opens the transform editor, which lets you complete the definition of the transform.

7. Enter option values.

To specify a data column as a transform option, enter the column name as it appears in the input schema or drag the column name from the input schema into the option box.

Related Information

18.2 Transform editors

After adding a transform to a data flow, you configure it using the transform editor.

Transform editor layouts vary based on the transform type.

❖ Example

The most commonly used transform is the Query transform. The Query transform editor has two panes:

- Either a [Schema In](#) or [Schema Out](#) area, or both.
- A tabbed options area where you set values based on the transform requirements and on the task to perform.

Data Quality transforms, such as Match and Data Cleanse, have a transform editor in which you set options and map input and output fields.

The Entity Extraction transform editor has settings for extraction options and an area to map input and output fields.

Refer to the *Reference Guide* for information about each transform editor.

18.3 Data Services Embedded Help

Embedded help is available for certain transforms in Designer.

The embedded help is the place to look when you need more information about Data Services transforms and options. The topic changes to help you with the context you're currently in. When you select a new transform or a new option group, the topic updates to reflect that selection.

You can also navigate to other topics by using hyperlinks within the open topic, or by using the table of contents.

The following transforms contain embedded help:

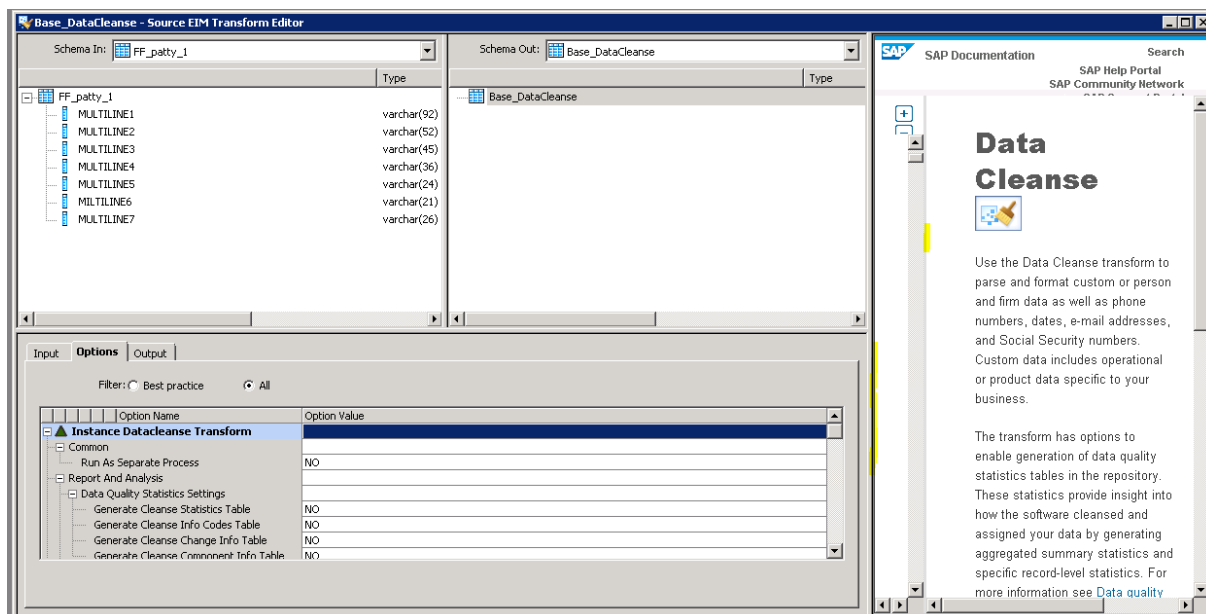
- Country ID
- Data Cleanse
- DQM Microservices
- Data Mask
- DSF2 Walk Sequencer
- Extraction
- Geocoder
- Global Address Cleanse
- Global Suggestion List
- Match Wizard
- USA Regulatory Address Cleanse
- Common

i Note

To view option information for the Associate, Match, and User Defined transforms:

1. Select the transform in a data flow.
2. Select **Tools** > **<transform> Editor**.

If the Help pane does not appear to the right of the transform editor pane, drag the right border of the editor to the left. (Shown highlighted in the following screen capture). To view or reduce the help table of contents, drag the border between the help pane and table of contents pane (also shown highlighted).



18.4 Transform configurations

A transform configuration is a transform with preconfigured best practice input fields, best practice output fields, and options that can be used in multiple data flows.

These configurations are useful if you repeatedly use a transform with specific options and input and output fields.

Some transforms, such as Data Quality transforms, have read-only transform configurations that the software installs when you install Data Services. You can create custom transform configurations, either by replicating an existing transform configuration or creating a new one. You cannot perform export or multi-user operations on read-only transform configurations.

In the *Transform Configuration Editor* dialog box, set up the default options, best practice input fields, and best practice output fields for your transform configuration. After you place an instance of the transform configuration in a data flow, you can override these preset defaults.

If you edit a transform configuration, that change is inherited by every instance of the transform configuration used in data flows, unless a user has explicitly overridden the same option value in an instance.

Related Information

[Creating a transform configuration \[page 304\]](#)

[Adding a user-defined field \[page 305\]](#)

18.4.1 Creating a transform configuration

Create your own transform configuration based on the common best practice input and output fields and common options that you use in your data flows.

1. Choose one of the following options based on how you want to create a configuration:
 - Right-click a transform configuration in the *Transforms* tab of the Local Object Library and select *New*.
 - Right-click a transform configuration in the *Transforms* tab of the Local Object Library and select *Replicate*.

You cannot create a transform configuration when the options *New* or *Replicate* are not available from the menu.

The *Transform Configuration Editor* window opens.

2. Enter a name for the new transform In *Transform Configuration Name*.
3. Open the *Options* tab and set the option values to determine how the transform processes your data.

If you are setting options for the Associate, Match, or User-Defined transforms, click *Edit Options*. The Associate Editor, Match Editor, or User-Defined Editor dialog box opens.

If you change an option value from its default value, a green triangle appears next to the option name to indicate that you made an override.

4. To designate an option as best practice, select the [Best Practice](#) checkbox next to the value.
A best practice indicator tells other users the typical options to set for this type of transform.
Use the filter to display all options or just those options that are designated as best practice options.
5. Click [Verify](#) to check that the selected option values are valid.
Errors display at the bottom of the window, if applicable.
6. In the [Input Best Practices](#) tab, select the input fields to designate as the best practice input fields for the transform configuration.
The provided transform configurations do not have best practice input fields. You select the best practice input fields based on the type of input schema that you prefer to use. For example, you may map the fields in your data flow that contain address data whether the address data resides in the following types of fields:
 - Discrete
 - Multiline
 - Discrete and multiline fields combinedWhen you use this transform configuration in a data flow, select the best practice filter in the [Input](#) tab of the transform editor. The software lists only the fields that you designated as best practice in the transform configuration.
7. Create user-defined input fields for the Associate, Match, and User-Defined transform configurations.
Create user-defined input fields by clicking the [Create](#) button.
8. In the [Output Best Practices](#) tab, select the output fields that you want to designate as the best practice output fields for the transform configuration.
When you use this transform configuration in a data flow, select the best practice filter in the [Output](#) tab of the transform editor. The software lists only the fields that you designated as best practice in the transform configuration.
9. Click [OK](#) to save the transform configuration.
The software lists the new transform configuration in the Local Object Library under the base transform of the same type.

Use the new transform configuration in data flows.

Related Information

[Reference Guide: Transforms, Transform configurations](#)

18.4.2 Adding a user-defined field

Transforms such as Associate, Match, and User-Defined can have user-defined fields because they do not have a predefined set of input fields.

Add a user-defined field to either a single instance of a transform in a data flow or to a transform configuration so that it can be used in all instances.

In the User-Defined transform, you can also add user-defined output fields.

1. In the Local Object Library, open the [Transforms](#) tab and right-click an existing Associate, Match, or User-Defined transform configuration and select [Edit](#).
The [Transform Configuration Editor](#) dialog box opens.
2. Open the [Input Best Practices](#) tab, click the [Create](#) button, and enter a name for the input field.
3. Click [OK](#) to save the transform configuration.

When you create a user-defined field in the transform configuration, Data Services displays it as an available field in each instance of the transform in a data flow. You can also create user-defined fields within each transform instance.

Related Information

18.4.3 Ordered options editor

Some transforms allow you to choose and specify the order of multiple values for a single option.

One example is the parser sequence option of the Data Cleanse transform.

To configure an ordered option:

1. Click the [Add](#) and [Remove](#) buttons to move option values between the Available and Selected values lists.

i Note

[Remove all values](#). To clear the Selected values list and move all option values to the Available values list, click [Remove All](#).

2. Select a value in the Available values list, and click the up and down arrow buttons to change the position of the value in the list.
3. Click [OK](#) to save your changes to the option configuration. The values are listed in the Designer and separated by pipe characters.

18.4.4 Associate, Match, and User-Defined transform editors

The Associate, Match, and User-Defined transform editors each have an additional editor to set detailed option information.

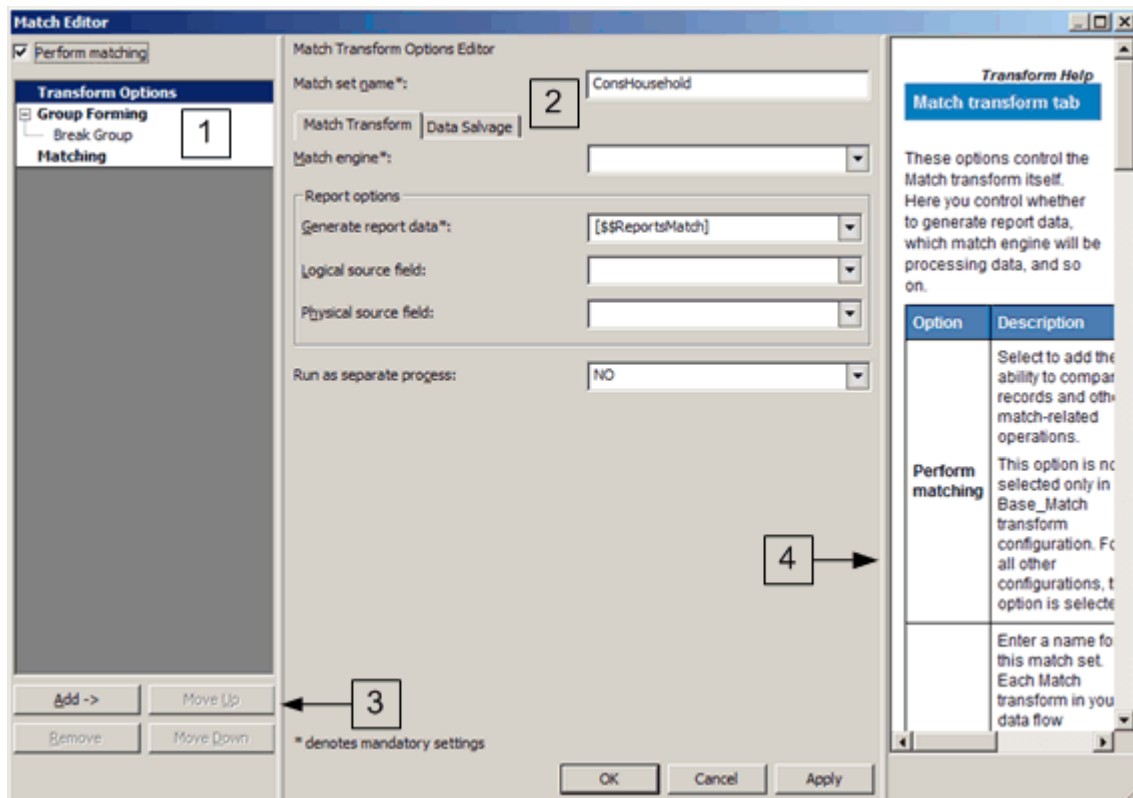
There are two ways to access the additional editor:

- Double-click the transform in the workspace to open its editor and then click [Edit Options](#).
- Select the transform in the data flow and select [Tools](#) > [<transform_name> Editor](#) [Tools](#) > [Associate Editor](#) [Tools](#) > [Match Editor](#) [Tools](#) > [User-Defined Editor](#).

Note

When you work with data flows that are created in Information Steward Data Cleansing Advisor, you cannot access the Match editor options. Match transform options cannot be edited; therefore, controls to access the Match editor are inactive.

The editors for the Associate, Match, and User-Defined transforms look and act similarly, and in some cases even share the same option groups.



The editor window is divided into four areas:

1. **Option Explorer** — In this area, you select the option groups, or operations, that are available for the transform. To display an option group that is hidden, right-click the option group it belongs to and select the name of the option group from the menu.
2. **Option Editor** — In this area, you specify the value of the option.
3. **Buttons** — Use these to add, remove and order option groups.
4. **Embedded help** — The embedded help displays additional information about using the current editor screen.

Related Information

[Reference Guide: Transforms, Associate](#)
[Reference Guide: Transforms, Match](#)

18.5 Data assessment

With operational systems frequently changing, data quality control becomes critical in your extract, transform, and load (ETL) jobs.

The software provides data quality controls that identify and fix errors in your data. These features can help ensure that you have trusted information.

The software provides the following features that you can use to determine and improve the quality and structure of your source data:

Feature	Description
Data Profiler	Use the Data Profiler to determine the following aspects of your data: • The quality of your source data before you extract it. The Data Profiler can identify anomalies in your source data to help you better define corrective actions in the validation transform, data quality, or other transforms. • The distribution, relationship, and structure of your source data to design your jobs and data flows better, and to design your target data warehouse. • The content of your source and target data so that you can verify that your data extraction job returns the results you expect.
View Data	Use the View Data feature to do the following: • View your source data before you execute a job to help you create higher quality job designs. • Compare sample data from different steps of your job to verify that your data extraction job returns the results you expect.
Design-Time Data Viewer	Use the Design-Time Data Viewer feature to view and analyze the input and output in real time as you design a transform. Use the Design-Time Data Viewer even before the data flow is complete or valid.
Validation transform	Use the validation transform to: • Verify that your source data meets your business rules. • Take appropriate actions when the data does not meet your business rules.
Auditing data flow	Use the auditing data flow feature to: • Define rules that determine if a source, transform, or target object processes correct data. • Define the actions to take when an audit rule fails.
Data quality transforms	Use data quality transforms to improve the quality of your data.

Feature	Description
Data validation dashboards	Use data validation dashboards in the metadata reporting tool to evaluate the reliability of your target data. Base reliability on the validation rules that you created in your batch jobs. This feedback allows business users to review, assess, and identify potential inconsistencies or errors in source data quickly.

Related Information

[Overview of data quality \[page 347\]](#)

[Using the Data Profiler \[page 309\]](#)

[Using View Data to determine data quality \[page 325\]](#)

[Using the Design-Time Data Viewer \[page 831\]](#)

[Using the Validation transform \[page 327\]](#)

[Using Auditing \[page 335\]](#)

[Management Console Guide: Data Validation Dashboard Reports](#)

[Reference Guide: Repository statistics tables for data quality, Data quality reports and statistics tables, Data quality statistics tables and supplemental content information](#)

18.5.1 Using the Data Profiler

The Data Profiler executes on a profiler server to provide the following data profiler information that multiple users can view:

Analysis	Description
Column analysis	The Data Profiler provides two types of column profiles: - Basic profiling—This information includes minimum value, maximum value, average value, minimum string length, and maximum string length. - Detailed profiling—Detailed column analysis includes distinct count, distinct percent, median, median string length, pattern count, and pattern percent.
Relationship analysis	This information identifies data mismatches between any two columns for which you define a relationship, including columns that have an existing primary key and foreign key relationship. You can save two levels of data: - Save the data only in the columns that you select for the relationship. - Save the values in all columns in each row.

18.5.1.1 Data sources that you can profile

SAP Data Services enables you to use the Data Profiler on many sources.

The following table lists the sources that you can use with the Data Profiler. See the Restrictions update in the latest *Release Notes* for a complete list of sources and targets that the Data Profiler supports.

Source/target	Includes
Databases	• Attunity Connector for mainframe databases • DB2 • Informix • Memory Datastore • MySQL • Microsoft SQL Server • Netezza • Oracle • ODBC • SAP HANA • SAP Sybase IQ • SAP Sybase SQL Anywhere • Siebel • Sybase ASE • Teradata
Applications	• Oracle E-Business Suite • People Tools • SAP Business Suite • SAP NetWeaver Business Warehouse • Siebel
Flat files	For example, plain text, binary file, comma-separated values and delimiter-separated values.
Tables	• SAP Business Suite • SAP BW Source

18.5.1.2 Connecting to the profiler server

Install and configure the profiler server before you use the Data Profiler.

The Designer must connect to the profiler server to run the Data Profiler and view the profiler results. You provide this connection information on the Profiler Server Login window.

1. Use one of the following methods to invoke the Profiler Server Login window:
 - From the tool bar menu, select  **Tools** > **Profiler Server Login** .
 - On the bottom status bar, double-click the Profiler Server icon which is to the right of the Job Server icon.
2. Enter your user credentials for the CMS.

Option	Description
System	Specify the server name and optionally the port for the CMS.
User name	Specify the user name to use to log into CMS.
Password	Specify the password to use to log into the CMS.
Authentication	Specify the authentication type used by the CMS.

- Click [Log on](#).
The software attempts to connect to the CMS using the specified information. When you log in successfully, the list of profiler repositories that are available to you is displayed.
- Select the repository you want to use.
- Click [OK](#) to connect using the selected repository.
When you successfully connect to the profiler server, the Profiler Server icon on the bottom status bar no longer has the red X on it. In addition, when you move the pointer over this icon, the status bar displays the location of the profiler server. Click

Related Information

[Management Console Guide: Profile Server Management](#)

[Administrator Guide: User and rights management](#)

18.5.1.3 Profiler statistics

18.5.1.3.1 Column profile

You can generate statistics for one or more columns.

The columns can all belong to one data source or from multiple data sources. If you generate statistics for multiple sources in one profile task, all sources must be in the same datastore.

Related Information

[Basic profiling \[page 312\]](#)

[Detailed profiling \[page 312\]](#)

[Examples \[page 313\]](#)

[Viewing the column attributes generated by the Data Profiler \[page 320\]](#)

[Submitting column profiler tasks \[page 314\]](#)

18.5.1.3.1.1 Basic profiling

By default, the Data Profiler generates basic profiler attributes for each column that you select.

The following table describes the basic profiler attributes for each attribute:

Basic attribute	Description
Min	Of all values, the lowest value in this column.
Min count	Number of rows that contain this lowest value in this column.
Max	Of all values, the highest value in this column.
Max count	Number of rows that contain this highest value in this column.
Average	For numeric columns, the average value in this column.
Min string length	For character columns, the length of the shortest string value in this column.
Max string length	For character columns, the length of the longest string value in this column.
Average string length	For character columns, the average length of the string values in this column.
Nulls	Number of NULL values in this column.
Nulls %	Percentage of rows that contain a NULL value in this column.
Zeros	Number of 0 values in this column.
Zeros %	Percentage of rows that contain a 0 value in this column.
Blanks	For character columns, the number of rows that contain a blank in this column.
Blanks %	Percentage of rows that contain a blank in this column.

18.5.1.3.1.2 Detailed profiling

Detailed attributes generation consumes more time and computer resources so they should be used for specific attributes.

You can generate more detailed attributes in addition to the basic attributes, but detailed attributes generation consumes more time and computer resources. Therefore, it is recommended that you do not select the detailed profile unless you need the following attributes:

Detailed attribute	Description
Median	The value that is in the middle row of the source table.
Median string length	For character columns, the value that is in the middle row of the source table.
Distincts	Number of distinct values in this column.
Distinct %	Percentage of rows that contain each distinct value in this column.
Patterns	Number of different patterns in this column.
Pattern %	Percentage of rows that contain each pattern in this column.

18.5.1.3.1.3 Examples

Examples of using column profile statistics to improve data quality

You can use the column profile attributes to assist you in different tasks, including the following tasks:

- Obtain basic statistics, frequencies, ranges, and outliers. For example, these profile statistics might show that a column value is markedly higher than the other values in a data source. You might then decide to define a validation transform to set a flag in a different table when you load this outlier into the target table.
- Identify variations of the same content. For example, part number might be an integer data type in one data source and a varchar data type in another data source. You might then decide which data type you want to use in your target data warehouse.
- Discover data patterns and formats. For example, the profile statistics might show that phone number has several different formats. With this profile information, you might decide to define a validation transform to convert them all to use the same target format.
- Analyze the numeric range. For example, customer number might have one range of numbers in one source, and a different range in another source. Your target will need to have a data type that can accommodate the maximum range.
- Identify missing information, nulls, and blanks in the source system. For example, the profile statistics might show that nulls occur for fax number. You might then decide to define a validation transform to replace the null value with a phrase such as "Unknown" in the target table.

18.5.1.3.2 Relationship profile

A relationship profile shows the percentage of non matching values in columns of two sources. The sources can be:

- Tables
- Flat files
- A combination of a table and a flat file

The key columns can have a primary key and foreign key relationship defined or they can be unrelated (as when one comes from a datastore and the other from a file format).

You can choose between two levels of relationship profiles to save:

Option	Description
Save key columns data only	By default, the Data Profiler saves the data only in the columns that you select for the relationship. i Note The Save key columns data only level is not available when using Oracle datastores.
Save all columns data	You can save the values in the other columns in each row, but this processing will take longer and consume more computer resources to complete.

When you view the relationship profile results, you can drill down to see the actual data that does not match.

You can use the relationship profile to assist you in different tasks, including the following tasks:

- Identify missing data in the source system. For example, one data source might include region, but another source might not.
- Identify redundant data across data sources. For example, duplicate names and addresses might exist between two sources or no name might exist for an address in one source.
- Validate relationships across data sources. For example, two different problem tracking systems might include a subset of common customer-reported problems, but some problems only exist in one system or the other.

Related Information

[Submitting relationship profiler tasks \[page 316\]](#)

[Viewing the profiler results \[page 320\]](#)

18.5.1.4 Executing a profiler task

The Data Profiler allows you to calculate profiler statistics for any set of columns you choose.

i Note

This optional feature is not available for columns with nested schemas, LONG or TEXT data type.

You cannot execute a column profile task with a relationship profile task.

18.5.1.4.1 Submitting column profiler tasks

1. In the Object Library of the Designer, you can select either a table or flat file.

For a table, go to the [Datastores](#) tab and select a table. If you want to profile all tables within a datastore, select the datastore name. To select a subset of tables in the [Datastores](#) tab, hold down the Ctrl key as you select each table.

For a flat file, go to the [Formats](#) tab and select a file.

2. After you select your data source, you can generate column profile statistics in one of the following ways:
 - Right-click and select [Submit Column Profile Request](#).
Some of the profile statistics can take a long time to calculate. Select this method so the profile task runs asynchronously and you can perform other Designer tasks while the profile task executes. This method also allows you to profile multiple sources in one profile task.
 - Right-click, select [View Data](#), click the [Profile](#) tab, and click [Update](#). This option submits a synchronous profile task, and you must wait for the task to complete before you can perform other tasks in the Designer.

You might want to use this option if you are already in the [View Data](#) window and you notice that either the profile statistics have not yet been generated, or the date that the profile statistics were generated is older than you want.

3. (Optional) Edit the profiler task name.

The Data Profiler generates a default name for each profiler task. You can edit the task name to create a more meaningful name, a unique name, or to remove dashes which are allowed in column names but not in task names.

If you select a single source, the default name has the following format:

`<username_t_sourceName>`

If you select multiple sources, the default name has the following format:

`<username_t_firstsourceName_lastsourceName>`

Column	Description
username	Name of the user that the software uses to access system services.
t	Type of profile. The value is C for column profile that obtains attributes (such as low value and high value) for each selected column.
firstsourceName	Name of first source in alphabetic order.
lastsourceName	Name of last source in alphabetic order if you select multiple sources.

4. If you select one source, the [Submit Column Profile Request](#) window lists the columns and data types.

Keep the check in front of each column that you want to profile and remove the check in front of each column that you do not want to profile.

Alternatively, you can click the check box at the top in front of [Name](#) to deselect all columns and then select the check boxes.

5. If you selected multiple sources, the [Submit Column Profiler Request](#) window lists the sources on the left.
 - a. Select a data source to display its columns on the right side.
 - b. On the right side of the [Submit Column Profile Request](#) window, keep the check in front of each column that you want to profile, and remove the check in front of each column that you do not want to profile.

Alternatively, you can click the check box at the top in front of [Name](#) to deselect all columns and then select the individual check box for the columns you want to profile.

- c. Repeat steps 1 and 2 for each data source.

6. (Optional) Select [Detailed profiling](#) for a column.

i Note

The Data Profiler consumes a large amount of resources when it generates detailed profile statistics. Choose Detailed profiling only if you want these attributes: distinct count, distinct percent, median value, median string length, pattern, pattern count. If you choose Detailed profiling, ensure that you specify a pageable cache directory that contains enough disk space for the amount of data you profile.

If you want detailed attributes for all columns in all sources listed, click [Detailed profiling](#) and select [Apply](#) to all columns of all sources.

If you want to remove Detailed profiling for all columns, click [Detailed profiling](#) and select [Remove](#) from all columns of all sources.

7. Click [Submit](#) to execute the profile task.

i Note

If the table metadata changed since you imported it (for example, a new column was added), you must re-import the source table before you execute the profile task.

If you clicked the [Submit Column Profile Request](#) option to reach this [Submit Column Profiler Request](#) window, the Profiler monitor pane appears automatically when you click [Submit](#).

If you clicked [Update](#) on the [Profile](#) tab of the [View Data](#) window, the [Profiler](#) monitor window does not appear when you click [Submit](#). Instead, a profile task is submitted asynchronously and you must wait for it to complete before you can do other tasks in the Designer .

You can also monitor your profiler task by name in the Administrator.

8. When the profiler task has completed, you can view the profile results in the [View Data](#) option.

Related Information

[Column profile \[page 311\]](#)

[Monitoring profiler tasks using the Designer \[page 319\]](#)

[Viewing the profiler results \[page 320\]](#)

[Administrator Guide: To configure run-time resources](#)

[Management Console Guide: Monitoring profiler tasks using the Administrator](#)

18.5.1.4.2 Submitting relationship profiler tasks

A relationship profile shows the percentage of non matching values in columns of two sources. The sources can be any of the following:

- Tables
- Flat files
- A combination of a table and a flat file

The columns can have a primary key and foreign key relationship defined or they can be unrelated (as when one comes from a datastore and the other from a file format).

The two columns do not need to be the same data type, but they must be convertible. For example, if you run a relationship profiler task on an integer column and a varchar column, the Data Profiler converts the integer value to a varchar value to make the comparison.

i Note

The Data Profiler consumes a large amount of resources when it generates relationship values. If you plan to use Relationship profiling, ensure that you specify a pageable cache directory that contains enough disk space for the amount of data you profile.

Related Information

[Data sources that you can profile \[page 310\]](#)

[Administrator Guide: Server Manager, Using the Server Manager on Windows, Configuring run-time resources](#)

18.5.1.4.2.1 Generating a relationship profile for columns in two sources

1. In the Object Library of the Designer, select two sources.

To select two sources in the same datastore or file format:

- a. Go to the [Datastore](#) or [Format](#) tab in the Object Library.
- b. Hold the Ctrl key down as you select the second table.
- c. Right-click and select [Submit Relationship Profile Request](#).

To select two sources from different datastores or files:

- a. Go to the [Datastore](#) or [Format](#) in the Object Library.
- b. Right-click the first source, select [Submit Relationship Profile Request](#) [Relationship with](#).
- c. Change to a different Datastore or Format in the Object Library
- d. Click the second source.

The [Submit Relationship Profile Request](#) window appears.

i Note

You cannot create a relationship profile for columns with a LONG or TEXT data type.

2. (Optional) Edit the profiler task name.

You can edit the task name to create a more meaningful name, a unique name, or to remove dashes, which are allowed in column names but not in task names. The default name that the Data Profiler generates for multiple sources has the following format:

`<username>_<t>_<firstsourcename>_<lastsourcename>`

Column	Description
username	Name of the user that the software uses to access system services.
t	Type of profile. The value is R for Relationship profile that obtains non matching values in the two selected columns.
firstsourcename	Name first selected source.
lastsourcename	Name last selected source.

3. By default, the upper pane of the [Submit Relationship Profile Request](#) window shows a line between the primary key column and foreign key column of the two sources, if the relationship exists. You can change the columns to profile.

The bottom half of the [Submit Relationship Profile Request](#) window shows that the profile task will use the equal (=) operation to compare the two columns. The Data Profiler will determine which values are not equal and calculate the percentage of non matching values.

4. To delete an existing relationship between two columns, select the line, right-click, and select [Delete Selected Relation](#).

To delete all existing relationships between the two sources, do one of the following actions:

- Right-click in the upper pane and click [Delete All Relations](#).
 - Click [Delete All Relations](#) near the bottom of the [Submit Relationship Profile Request](#) window.
5. If a primary key and foreign key relationship does not exist between the two data sources, specify the columns to profile. You can resize each data source to show all columns.

To specify or change the columns for which you want to see relationship values:

- a. Move the cursor to the first column to select. Hold down the cursor and draw a line to the other column that you want to select.
 - b. If you deleted all relations and you want the Data Profiler to select an existing primary-key and foreign-key relationship, either right-click in the upper pane and click [Propose Relation](#), or click [Propose Relation](#) near the bottom of the [Submit Relationship Profile Request](#) window.
6. By default, the [Save key columns data only](#) option is selected. This option indicates that the Data Profiler saves the data only in the columns that you select for the relationship, and you will not see any sample data in the other columns when you view the relationship profile.

If you want to see values in the other columns in the relationship profile, select the [Save all columns data](#) option.

7. Click [Submit](#) to execute the profiler task.

i Note

If the table metadata changed since you imported it (for example, a new column was added), you must re-import the source table before you execute the profile task.

8. The Profiler monitor pane appears automatically when you click [Submit](#).

You can also monitor your profiler task by name in the Administrator.

9. When the profiler task has completed, you can view the profile results in the [View Data](#) option when you right click on a table in the Object Library.

Related Information

[Viewing the relationship profile data generated by the Data Profiler \[page 324\]](#)

[Monitoring profiler tasks using the Designer \[page 319\]](#)

[Viewing the profiler results \[page 320\]](#)

18.5.1.5 Monitoring profiler tasks using the Designer

The [Profiler](#) monitor window appears automatically when you submit a profiler task if you clicked the menu bar to view the [Profiler](#) monitor window. You can dock this profiler monitor pane in the Designer or keep it separate.

The Profiler monitor pane displays the currently running task and all of the profiler tasks that have executed within a configured number of days.

- Click the [Refresh](#)
- Click the [Information](#) icon to view the sources that the selected task is profiling. If the task failed, the Information window also displays an error message.

The Profiler monitor shows the following columns: The icons in the upper-left corner of the Profiler monitor display the following information:

Column	Description
Name	Name of the profiler task that was submitted from the Designer. The icons in the upper-left corner of the Profiler If the profiler task is for a single source, the default name has the following format: <pre><username_t_sourcename></pre> If the profiler task is for multiple sources, the default name has the following format: <pre><username_t_firstsourcename_lastsourcename></pre>
Type	The type of profiler task can be: • icon to refresh the Profiler monitor pane and display the latest status of profiler tasks. Column icon to refresh the Profiler monitor pane and display the latest status of profiler tasks. • Relationship
Status	The status of a profiler task can be: • Done—The task completed successfully. • Pending—The task is on the wait queue because the maximum number of concurrent tasks has been reached or another task is profiling the same table. • Running—The task is currently executing. • Error—The task terminated with an error. Double-click on the value in this Status column to display the error message.
Timestamp	Date and time that the profiler task executed.
Sources	Names of the tables for which the profiler task executes.

Related Information

[Executing a profiler task \[page 314\]](#)

18.5.1.6 Viewing the profiler results

The Data Profiler calculates and saves the profiler attributes into a profiler repository that multiple users can view.

Related Information

[Viewing the column attributes generated by the Data Profiler \[page 320\]](#)

[Viewing the relationship profile data generated by the Data Profiler \[page 324\]](#)

18.5.1.6.1 Viewing the column attributes generated by the Data Profiler

1. In the Object Library, select the table for which you want to view profiler attributes.
2. Right-click and select [View Data](#).
3. Click the [Profile](#) tab (second) to view the column profile attributes.
 - a. The [Profile](#) tab shows the number of physical records that the Data Profiler processed to generate the values in the profile grid.
 - b. The profile grid contains the column names in the current source and profile attributes for each column. To populate the profile grid, execute a profiler task or select names from this column and click [Update](#).
 - c. You can sort the values in each attribute column by clicking the column heading. The value *n/a* in the profile grid indicates an attribute does not apply to a data type,

Basic profile attribute	Description	Relevant data type		
		Character	Numeric	Datetime
Min	Of all values, the lowest value in this column.	Yes	Yes	Yes
Min count	Number of rows that contain this lowest value in this column.	Yes	Yes	Yes
Max	Of all values, the highest value in this column.	Yes	Yes	Yes

Basic profile attribute	Description	Relevant data type		
		Character	Numeric	Datetime
Max count	Number of rows that contain this highest value in this column.	Yes	Yes	Yes
Average	For numeric columns, the average value in this column.	n/a	Yes	n/a
Min string length	For character columns, the length of the shortest string value in this column.	Yes	No	No
Max string length	For character columns, the length of the longest string value in this column.	Yes	No	No
Average string length	For character columns, the average length of the string values in this column.	Yes	No	No
Nulls	Number of NULL values in this column.	Yes	Yes	Yes
Nulls %	Percentage of rows that contain a NULL value in this column.	Yes	Yes	Yes
Zeros	Number of 0 values in this column.	No	Yes	No
Zeros %	Percentage of rows that contain a 0 value in this column.	No	Yes	No
Blanks	For character columns, the number of rows that contain a blank in this column.	Yes	No	No
Blanks %	Percentage of rows that contain a blank in this column.	Yes	No	No

- d. If you selected the [Detailed profiling](#) option on the [Submit Column Profile Request](#) window, the [Profile](#) tab also displays the following detailed attribute columns.

Detailed profile attribute	Description	Relevant data type		
		Character	Numeric	Datetime
Distincts	Number of distinct values in this column.	Yes	Yes	Yes
Distinct %	Percentage of rows that contain each distinct value in this column.	Yes	Yes	Yes
Median	The value that is in the middle row of the source table.	Yes	Yes	Yes
Median string length	For character columns, the value that is in the middle row of the source table.	Yes	No	No
Pattern %	Percentage of rows that contain each distinct value in this column. The format of each unique pattern in this column.	Yes	No	No
Patterns	Number of different patterns in this column.	Yes	No	No

- Click an attribute value to view the entire row in the source table. The bottom half of the [View Data](#) window displays the rows that contain the attribute value that you clicked. You can hide columns that you do not want to view by clicking the Show/Hide Columns icon.

For example, your target ADDRESS column might only be 45 characters, but the Profiling data for this Customer source table shows that the maximum string length is 46. Click the value 46 to view the actual data. You can resize the width of the column to display the entire string.

- (Optional) Click [Update](#) if you want to update the profile attributes. Reasons to update at this point include:
 - The profile attributes have not yet been generated.
 - The date that the profile attributes were generated is older than you want. The Last updated value in the bottom left corner of the [Profile](#) tab is the timestamp when the profile attributes were last generated.

i Note

The Update option submits a synchronous profiler task, and you must wait for the task to complete before you can perform other tasks in the Designer.

The [Submit column Profile Request](#) window appears.

Select only the column names you need for this profiling operation because Update calculations impact performance. You can also click the check box at the top in front of Name to deselect all columns and then select each check box in front of each column you want to profile.

- Click a statistic in either Distincts or Patterns to display the percentage of each distinct value or pattern value in a column. The pattern values, number of records for each pattern value, and percentages appear on the right side of the *Profile* tab.

For example, the following *Profile* tab for table CUSTOMERS shows the profile attributes for column REGION. The Distincts attribute for the REGION column shows the statistic 19 which means 19 distinct values for REGION exist.

View Data... - CUSTOMERS(Northwind_DS.DBO)

Update Records : 91

Column	Median string length	Distincts	Nulls	Nulls %	Zeros	Zeros %	Blanks	Blanks %
CUSTOMERID	5	91	0	0%	n/a	n/a	0	0%
COMPANYNAME	18	91	0	0%	n/a	n/a	0	0%
CONTACTNAME	14	91	0	0%	n/a	n/a	0	0%
CONTACTTITLE	17	12	0	0%	n/a	n/a	0	0%
ADDRESS	18	91	0	0%	n/a	n/a	0	0%
CITY	8	69	0	0%	n/a	n/a	0	0%
REGION	2	19	60	65.93%	n/a	n/a	0	0%
POSTALCODE	5	87	1	1.09%	n/a	n/a	0	0%
COUNTRY	6	21	0	0%	n/a	n/a	0	0%
PHONE	12	91	0	0%	n/a	n/a	0	0%
FAX	12	70	22	24.17%	n/a	n/a	0	0%

Profiling data (REGION,Distincts)

Value	Records	% of total
<Null>	60	65.93%
SP	6	6.59%
OR	4	4.40%
WA	3	3.30%
RJ	3	3.30%
BC	2	2.20%
Táchira	1	1.10%
Québec	1	1.10%
Nueva Esparta	1	1.10%
NM	1	1.10%
MT	1	1.10%

Profiling data (<Null>,Value)

CUSTOMERID	COMPANYNAME	CONTACTNAME	CONTACTTITLE	ADDRESS	CITY	REGION
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	<Null>
ANATR	Ana Trujillo Emparedados y...	Ana Trujillo	Owner	Avda. de la Constitución 2...	México D.F.	<Null>
ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312	México D.F.	<Null>

Record: 1 of 60
Last updated: 11/21/06 11:47:00 AM

- Click the statistic in the Distincts column to display each of the 19 values and the percentage of rows in table CUSTOMERS that have that value for column REGION. In addition, the bars in the right-most column show the relative size of each percentage.
- The Profiling data on the right side shows that a very large percentage of values for REGION is Null. Click either Null under Value or 60 under Records to display the other columns in the rows that have a Null value in the REGION column.
- Your business rules might dictate that REGION should not contain Null values in your target data warehouse. Therefore, decide what value you want to substitute for Null values when you define a validation transform.

Related Information

[Executing a profiler task \[page 314\]](#)

[Defining a validation rule based on a column profile \[page 329\]](#)

18.5.1.6.2 Viewing the relationship profile data generated by the Data Profiler

Relationship profile data shows the percentage of non matching values in columns of two sources. The sources can be tables, flat files, or a combination of a table and a flat file. The columns can have a primary key and foreign key relationship defined or they can be unrelated (as when one comes from a datastore and the other from a file format).

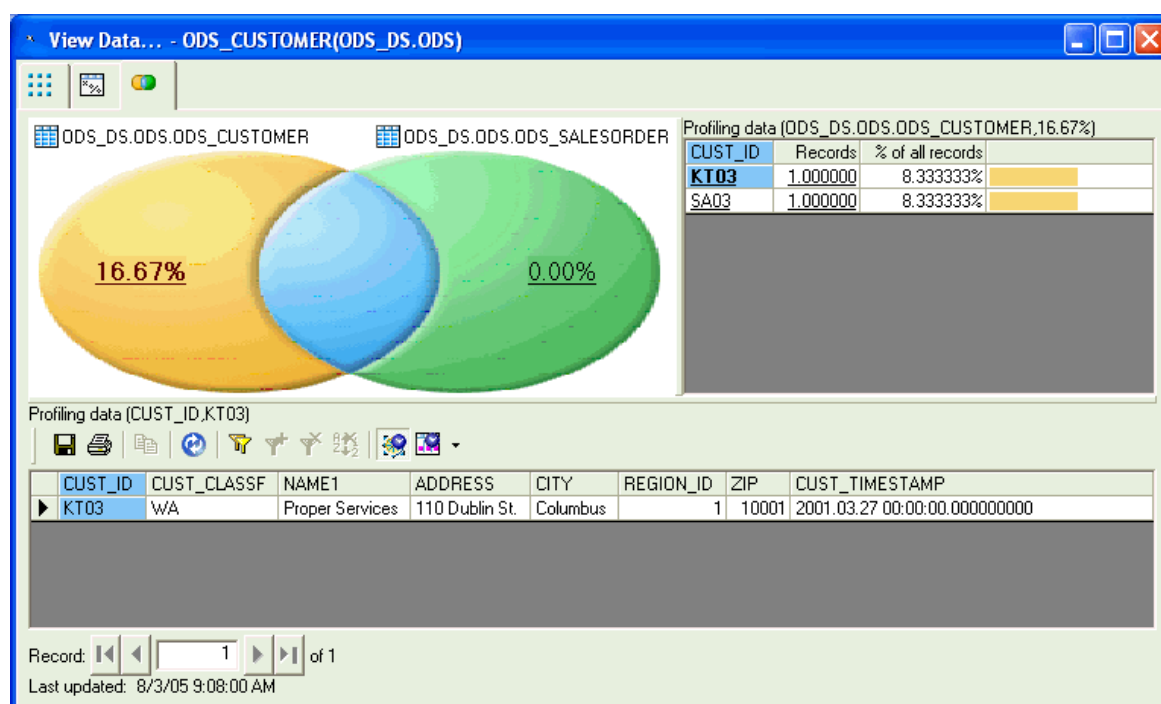
1. In the Object Library, select the table or file for which you want to view relationship profile data.
2. Right-click and select *View Data*.
3. Click the *Relationship* tab (third) to view the relationship profile results.

i Note

The *Relationship* tab is visible only if you executed a relationship profiler task.

4. Click the nonzero percentage in the diagram to view the key values that are not contained within the other table.

For example, the following View Data Relationship tab shows the percentage (16.67) of customers that do not have a sales order. The relationship profile was defined on the CUST_ID column in table ODS_CUSTOMER and CUST_ID column in table ODS_SALESORDER. The value in the left oval indicates that 16.67% of rows in table ODS_CUSTOMER have CUST_ID values that do not exist in table ODS_SALESORDER.



Click the 16.67 percentage in the ODS_CUSTOMER oval to display the CUST_ID values that do not exist in the ODS_SALESORDER table. The non matching values KT03 and SA03 display on the right side of the Relationship tab. Each row displays a non matching CUST_ID value, the number of records with that CUST_ID value, and the percentage of total customers with this CUST_ID value.

5. Click one of the values on the right side to display the other columns in the rows that contain that value.

The bottom half of the [Relationship Profile](#) tab displays the values in the other columns of the row that has the value KT03 in the column CUST_ID.

i Note

If you did not select Save all column data on the [Submit Relationship Profile Request](#) window, you cannot view the data in the other columns.

Related Information

[Submitting relationship profiler tasks \[page 316\]](#)

18.5.2 Using View Data to determine data quality

Use View Data to help you determine the quality of your source and target data. View Data provides the capability to:

- View sample source data before you execute a job to create higher quality job designs.
- Compare sample data from different steps of your job to verify that your data extraction job returns the results you expect.

i Note

To use View Data on a file, the file must be available from the Designer.

Related Information

[Defining a validation rule based on a column profile \[page 329\]](#)

[Using View Data \[page 819\]](#)

18.5.2.1 Data tab

The [Data](#) tab is always available and displays the data contents of sample rows. You can display a subset of columns in each row and define filters to display a subset of rows.

For example, your business rules might dictate that all phone and fax numbers be in one format for each country. The following [Data](#) tab shows a subset of rows for the customers that are in France.

View Data... - CUSTOMERS(Northwind_DS.DBO)

CUSTOMERID	COMPANYNAME	CITY	POSTALCODE	COUNTRY	PHONE	FAX
BLONP	BlondesddsI père et fils	Strasbourg	67000	France	88.60.15.31	88.60.15.32
BONAP	Bon app'	Marseille	13008	France	91.24.45.40	91.24.45.41
DUMON	Du monde entier	Nantes	44000	France	40.67.88.88	40.67.89.89
FOLIG	Folies gourmandes	Lille	59000	France	20.16.10.16	20.16.10.17
FRANR	France restauration	Nantes	44000	France	40.32.21.21	40.32.21.20
LACDR	La corne d'abondance	Versailles	78000	France	30.59.84.10	30.59.85.11
LAMAI	La maison d'Asie	Toulouse	31000	France	61.77.61.10	61.77.61.11
PARIS	Paris spécialités	Paris	75012	France	(1) 42.34.22.66	(1) 42.34.22.77
SPECD	Spécialités du monde	Paris	75016	France	(1) 47.55.60.10	(1) 47.55.60.20
VICTE	Victuailles en stock	Lyon	69004	France	78.32.54.86	78.32.54.87
VINET	Vins et alcools Chevalier	Reims	51100	France	26.47.15.10	26.47.15.11

Record: 1 of 11 (filtered: (COUNTRY = 'France'))

Notice that the PHONE and FAX columns displays values with two different formats. You can now decide which format you want to use in your target data warehouse and define a validation transform accordingly.

Related Information

[View Data Properties \[page 823\]](#)

[Defining a validation rule based on a column profile \[page 329\]](#)

[Data tab \[page 827\]](#)

18.5.2.2 Profile tab

Two displays are available on the *Profile* tab:

- Without the Data Profiler, the *Profile* tab displays the following column attributes: distinct values, NULLs, minimum value, and maximum value.
- If you configured and use the Data Profiler, the *Profile* tab displays the same above column attributes plus many more calculated statistics, such as average value, minimum string length, and maximum string length, distinct count, distinct percent, median, median string length, pattern count, and pattern percent.

Related Information

[Profile tab \[page 828\]](#)

[Viewing the column attributes generated by the Data Profiler \[page 320\]](#)

18.5.2.3 Relationship Profile or Column Profile tab

The third tab that displays depends on whether or not you configured and use the Data Profiler.

- If you do not use the Data Profiler, the [Column Profile](#) tab allows you to calculate statistical information for a single column.
- If you use the Data Profiler, the [Relationship](#) tab displays the data mismatches between two columns from which you can determine the integrity of your data between two sources.

Related Information

[Column Profile tab \[page 829\]](#)

[Viewing the relationship profile data generated by the Data Profiler \[page 324\]](#)

18.5.3 Using the Validation transform

The Data Profiler and View Data features can identify anomalies in incoming data. You can then use a Validation transform to define the rules that sort good data from bad. You can write the bad data to a table or file for subsequent review.

For details on the Validation transform including how to implement reusable validation functions, see the *SAP Data Services Reference Guide*.

Related Information

[Reference Guide: Transforms, Platform transforms, Validation](#)

18.5.3.1 Analyzing the column profile

You can obtain column profile information by submitting column profiler tasks.

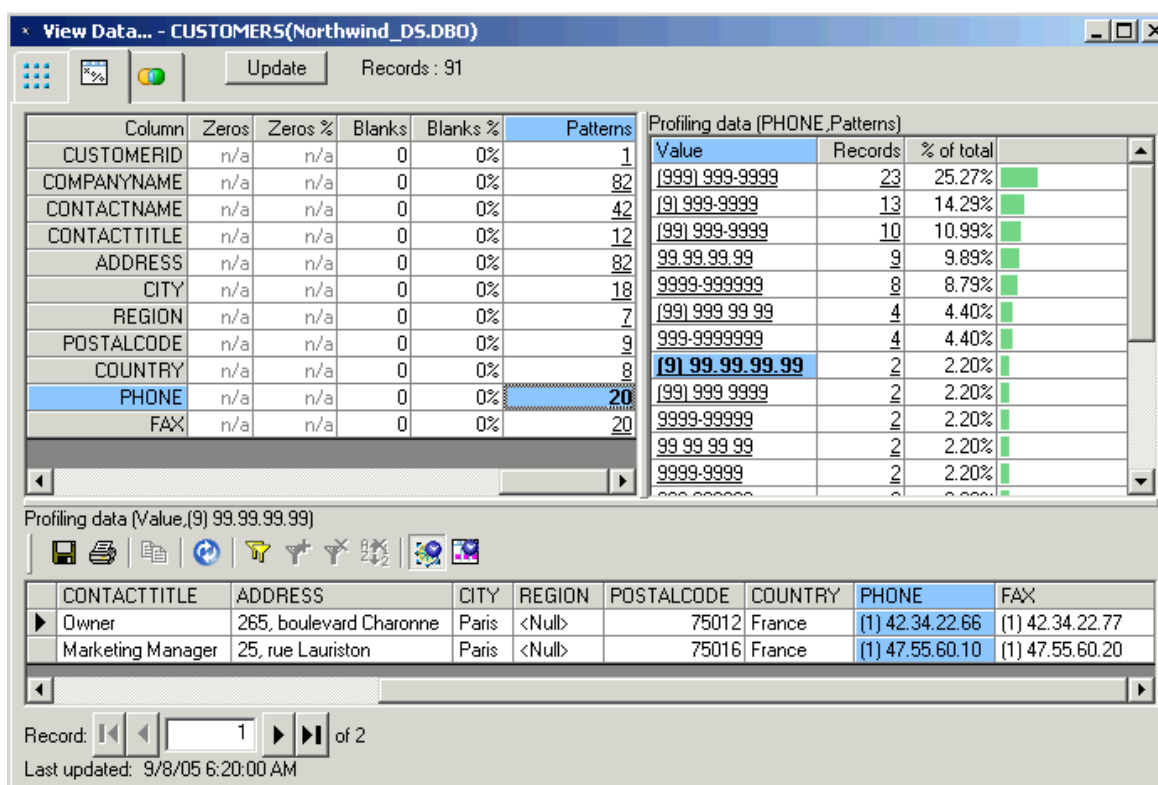
For example, suppose you want to analyze the data in the Customers table in the Microsoft SQL Server Northwinds sample database.

Related Information

[Submitting column profiler tasks \[page 314\]](#)

18.5.3.1.1 Analyzing column profile attributes

1. In the object library, right-click the profiled Customers table and select [View Data](#).
2. Select the [Profile](#) tab in the [View Data](#) window. The Profile tab displays the column-profile attributes shown in the following graphic.



The Patterns attribute for the PHONE column shows the value 20, which means 20 different patterns exist.

3. Click the value 20 in the [Patterns](#) attribute column. The [Profiling data](#) pane displays the individual patterns for the column PHONE and the percentage of rows for each pattern.
4. Suppose that your business rules dictate that all phone numbers in France should have the format 99.99.99.99. However, the profiling data shows that two records have the format (9) 99.99.99.99. To display the columns for these two records in the bottom pane, click either (9) 99.99.99.99 under Value or click 2 under Records. You can see that some phone numbers in France have a prefix of (1).

You can use a Validation transform to identify rows containing the unwanted prefix. Then you can correct the data to conform to your business rules then reload it.

The next section describes how to configure the Validation transform to identify the errant rows.

Related Information

[Defining a validation rule based on a column profile \[page 329\]](#)

18.5.3.2 Defining a validation rule based on a column profile

This section takes the Data Profiler results and defines the Validation transform according to the sample business rules. Based on the preceding example of the phone prefix (1) for phone numbers in France, the following procedure describes how to define a data flow and validation rule that identifies that pattern. You can then review the failed data, make corrections, and reload the data.

18.5.3.2.1 Defining the validation rule that identifies a pattern

This procedure describes how to define a data flow and validation rule that identifies rows containing the (1) prefix described in the previous section.

1. Create a data flow with the Customers table as a source, add a Validation transform and a target, and connect the objects.
2. Open the Validation transform by clicking its name.
3. In the transform editor, click [Add](#).
The Rule Editor dialog box displays.
4. Type a [Name](#) and optionally a [Description](#) for the rule.
5. Verify the [Enabled](#) check box is selected.
6. For [Action on Fail](#), select [Send to Fail](#).
7. Select the [Column Validation](#) radio button.
 - a. Select the [Column](#) CUSTOMERS.PHONE from the drop-down list.
 - b. For [Condition](#), from the drop-down list select [Match pattern](#).
 - c. For the value, type the expression '99.99.99.99'.
8. Click [OK](#).
The rule appears in the Rules list.

After running the job, the incorrectly formatted rows appear in the Fail output. You can now review the failed data, make corrections as necessary upstream, and reload the data.

Related Information

[Analyzing the column profile \[page 327\]](#)

18.5.3.3 Validation transform Rule Editor

Use the [Rule Editor](#) in the Validation transform to define or edit a validation rule.

Launch the Rule Editor from the Validation Rules tab of the Validation transform by clicking [Add](#) or double-clicking an existing rule to edit it.

The [Rule Editor](#) lets you configure a rule based on an existing validation function or a rule defined in the Validation transform based on a single column input.

Rules defined with a validation function have the following characteristics:

- Reusable.
- Accept multiple input parameters.
- Used for more complex rules.
- Can be created in SAP Information Steward and imported and used in Data Services.

Rules defined with a column validation have the following characteristics:

- Not reusable. The rule definition is part of the transform and you cannot share it outside the transform.
- Binds to only one input column.
- Used for simple rules.

There are two types of validation functions as categorized in the object library:

- Imported from Information Steward: Created in Information Steward and cannot be edited in Data Services.
- Locally created: Created and editable in Data Services.

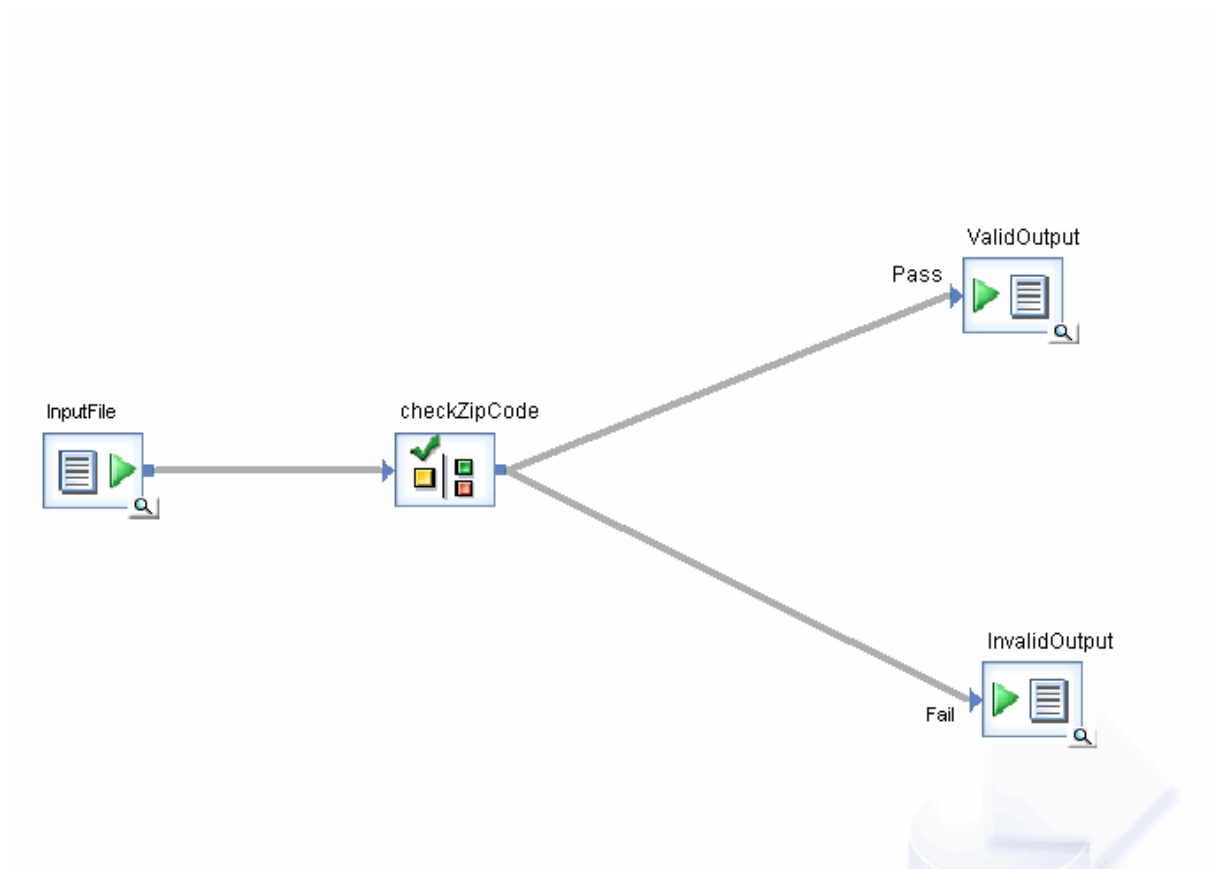
18.5.3.3.1 Defining a validation rule based on a column

Use the Validation transform to create a validation rule based on a column in your data.

1. In a data flow with the Validation transform connected to an input schema, click the transform name to open its editor.
2. On the Validation Rules tab, click [Add](#).
The Rule Editor displays with the [Enabled](#) check box selected by default.
3. Type a name for the rule and optionally add a description.
4. Select an [Action on fail](#):
 - Send To Fail
 - Send To Pass
 - Send To Both
5. Select [Column Validation](#).
6. Select a [Column](#) from the drop-down list.
7. Define a condition. All conditions must be Boolean expressions. Or, select [Custom Condition](#) to enable access to the smart editor (ellipses button) or function wizard ([Functions](#) button).

Example: Defining a validation rule for a five-digit ZIP code pattern

The following example defines a Validation transform that verifies that the data for ZIP code is a five-digit number.



1. Open the Validation transform editor.
2. Click [Add](#).
3. Name the rule **checkzip**.
4. For *Action on Fail*, select *Send To Both*.
5. Select *Column Validation*.
6. Select the *Customers.Zipcode* column.
7. Select the *LIKE* condition.
8. Specify the pattern as **99999**, which matches any five-digit number.
9. Click *OK*.
10. On the *Validation Rules* tab, for the option, *If any rule fails and send to Pass, substitute with:* double-click a cell under *Column* and select *Customers.Zipcode*.
11. Double-click in the cell under *Expression* and type '**Invalid**'.

If the zip code does not match the pattern, Data Services replaces the value with the string **Invalid**.

The following graphic shows the input data.

Q "C:\work\validation_data\demo\orderinfo2.txt"(OrderInfo)

OrderNum	OrderDate	Name	Address1	Address2	City	State	Zipcode	Phone
5001	2004.05.12	First National bank	1 First Street	NULL	San Jose	CA	95134	408-933-6821
5002	NULL	Second National bank	123 First Street	NULL	San Jose	CA	95134	408-933-3456
3500	2004.05.01	Second National bank	2 First Street	NULL	San Jose	CA	95134	408-933-6821
5003	2004.05.12	First National bank	1 First Street	NULL	San Jose	CA	NULL	408-933-6821
1	2004.06.12	First National bank	123 First Street	NULL	San Jose	CA	95134	408-933-6821
5005	NULL	Third National bank	234 Commercial Street	NULL	San Francisco	CA	95100	650-933-3456
5006	2004.05.16	First National bank	1 First Street	NULL	San Jose	CA	95134	408-933-6821
5007	2004.05.16	First National bank	1 First Street	NULL	San Jose	CA	951ab	408-933-6821
5008	2004.06.18	Third National bank	234 Commercial Street	NULL	San Francisco	CA	95100	650-933-3456
5009	NULL	Third National bank	234 Commercial Street	NULL	San Francisco	CA	95100	650-933-3456
5007	2004.05.16	First National bank	1 First Street	NULL	San Jose	CA	1234567	408-933-6821

The following graphic shows the data sent to the Pass/ValidOutput table:

Q "C:\work\validation_data\demo\orderinfo2_out.txt"(OrderInfo)

OrderNum	OrderDate	Name	Address1	Address2	City	State	Zipcode	Phone
5001	2004.05.12	First National bank	1 First Street	NULL	San Jose	CA	95134	408-933-6821
5002	NULL	Second National bank	123 First Street	NULL	San Jose	CA	95134	408-933-3456
3500	2004.05.01	Second National bank	2 First Street	NULL	San Jose	CA	95134	408-933-6821
5003	2004.05.12	First National bank	1 First Street	NULL	San Jose	CA	Invalid	408-933-6821
1	2004.06.12	First National bank	123 First Street	NULL	San Jose	CA	95134	408-933-6821
5005	NULL	Third National bank	234 Commercial Street	NULL	San Francisco	CA	95100	650-933-3456
5006	2004.05.16	First National bank	1 First Street	NULL	San Jose	CA	95134	408-933-6821
5007	2004.05.16	First National bank	1 First Street	NULL	San Jose	CA	Invalid	408-933-6821
5008	2004.06.18	Third National bank	234 Commercial Street	NULL	San Francisco	CA	95100	650-933-3456
5009	NULL	Third National bank	234 Commercial Street	NULL	San Francisco	CA	95100	650-933-3456
5007	2004.05.16	First National bank	1 First Street	NULL	San Jose	CA	Invalid	408-933-6821

The following graphic shows the data sent to the Fail/InvalidOutput table:

Q "C:\work\validation_data\demo\orderinfo2_failed.txt"(FailedOrderInfo)

Ord...	OrderDate	Name	Address1	Addr...	City	State	Zipcode	Phone	D	DI_ERRORCOLUMNS
5003	2004.05.12	First National bank	1 First Street	NULL	San Jose	CA	NULL	408-933-6821	B	checkZipCode failed rule(s): Zipcode
5007	2004.05.16	First National bank	1 First Street	NULL	San Jose	CA	951ab	408-933-6821	B	checkZipCode failed rule(s): Zipcode
5007	2004.05.16	First National bank	1 First Street	NULL	San Jose	CA	1234567	408-933-6821	B	checkZipCode failed rule(s): Zipcode

18.5.3.3.2 Define validation rules with a custom validation function

Use a custom validation function to define a validation rule.

To create a validation rule using a custom validation function, perform two groups of steps.

1. [Defining a custom validation function in the object library \[page 333\]](#)
Create a validation function in the Data Services object library to use in data flows.
2. [Defining a validation rule using a validation function \[page 334\]](#)
Define a validation rule in the Validation transform using a custom function that you create in Data Services object library.

18.5.3.3.2.1 Defining a custom validation function in the object library

Create a validation function in the Data Services object library to use in data flows.

This procedure uses an example to describe how to add and define a custom validation function in the object library.

❖ Example

In this example, the function determines whether the ZIP column contains 5-digit ZIP codes by checking each character and ensuring that each character is a digit. To verify the character is a digit, the function checks to see if its ASCII value is between 48 and 57 inclusive (corresponding to 0 through 9 respectively).

1. From the *Custom Function* tab of the object library, right-click *Validation Functions* and select *New*. The *Custom Function* dialog box displays with *Validation function* selected.
2. Enter the name **zipIsValid** for the new function and click *Next*. The smart editor displays.
3. To define the parameters and variables for this function, select the *Variables* tab in the smart editor window.
4. Right-click the *Parameters* node and select *Insert*.
5. Enter a parameter name such as **\$ZipToValidate**.
6. Select the appropriate data type such as *varchar* with a length of **5**.
7. For the *Parameter type* select *Input*.

i Note

Validation functions can have multiple input parameters; additional output parameters are not used.

8. Click *OK*.
The parameter \$ZipToValidate appears under the *Parameters* node.
9. Add five local variables named \$Z1 through \$Z5, which represent the five characters in the column. Right-click *Local*, select *Insert*, enter the name, select the data type *int*, and click *OK*.
The five variables appear under the *Local* variables node.
10. In the text panel of the smart editor, enter the following validation script:

```
$Z1 = ascii(substr($ZipToValidate, 1, 1));
$Z2 = ascii(substr($ZipToValidate, 2, 1));
$Z3 = ascii(substr($ZipToValidate, 3, 1));
$Z4 = ascii(substr($ZipToValidate, 4, 1));
$Z5 = ascii(substr($ZipToValidate, 5, 1));
if ( $Z1 < 48 OR $Z1 > 57 )
    return 0;
if ( $Z2 < 48 OR $Z2 > 57 )
```

```

return 0;
if ( $Z3 < 48 OR $Z3 > 57 )
    return 0;
if ( $Z4 < 48 OR $Z4 > 57 )
    return 0;
if ( $Z5 < 48 OR $Z5 > 57 )
    return 0;
return 1;

```

i Note

The Validation transform only supports custom functions that return an integer data type. If a return value is not a zero, then Data Services processes it as TRUE.

11. Click [OK](#).

The new function displays in the object library under the [Validation Function](#) node.

Continue to the next procedure to use the validation function in a validation rule.

Task overview: [Define validation rules with a custom validation function \[page 332\]](#)

Next task: [Defining a validation rule using a validation function \[page 334\]](#)

18.5.3.3.2 Defining a validation rule using a validation function

Define a validation rule in the Validation transform using a custom function that you create in Data Services object library.

1. Add a data flow with a source and a Validation transform and connect them.
2. Click the transform name to open the Validation transform editor.
3. On the [Validation Rules](#) tab, click [Add](#).
The Rule Editor displays with the [Enabled](#) check box selected by default.
4. In the Rule Editor, name the rule.
5. Select the [Action on fail: Send to Fail](#).
6. With the [Validation Function](#) option selected, from the drop-down list select the `ZipIsValid` function.
The Bindings table populates with the required parameters for the function.
7. Define the parameter argument. For the `$ZipToValidate` parameter, double-click the cell under [Argument](#) and select the column [ZIP](#).
8. Click [OK](#) to save the rule.
The rule appears in the Rules list on the [Validation Rules](#) tab.

When you run the job, the rule will apply the reusable validation function to the ZIP column and check each character to validate each is a digit.

Task overview: [Define validation rules with a custom validation function \[page 332\]](#)

Previous task: [Defining a custom validation function in the object library \[page 333\]](#)

18.5.4 Using Auditing

Auditing provides a way to ensure that a data flow loads correct data into the warehouse. Use auditing to perform the following tasks:

- Define audit points to collect run time statistics about the data that flows out of objects. Auditing stores these statistics in the repository.
- Define rules with these audit statistics to ensure that the data at the following points in a data flow is what you expect:
 - Extracted from sources
 - Processed by transforms
 - Loaded into targets
- Generate a run time notification that includes the audit rule that failed and the values of the audit statistics at the time of failure.
- Display the audit statistics after the job execution to help identify the object in the data flow that might have produced incorrect data.

i Note

If you add an audit point prior to an operation that is usually pushed down to the database server, performance might degrade because push-down operations cannot occur after an audit point.

18.5.4.1 Auditing objects in a data flow

You can collect audit statistics on the data that flows out of any object, such as a source, transform, or target. If a transform has multiple distinct or different outputs (such as Validation or Case), you can audit each output independently.

To use auditing, you define the following objects in the *Audit* window:

Object name	Description
Audit point	The object in a data flow where you collect audit statistics. You can audit a source, a transform, or a target. You identify the object to audit when you define an audit function on it.

Object name	Description															
Audit function	The audit statistic that the software collects for a table, output schema, or column. The following table shows the audit functions that you can define.															
	<table><tr><th>Data object</th><th>Audit function</th><th>Description</th></tr><tr><td>Table or output schema</td><td>Count</td><td>This function collects two statistics: • Good count for rows that were successfully processed. • Error count for rows that generated some type of error if you enabled error handling. </td></tr><tr><td>Column</td><td>Sum</td><td>Sum of the numeric values in the column. Applicable data types include decimal, double, integer, and real. This function only includes the Good rows.</td></tr><tr><td>Column</td><td>Average</td><td>Average of the numeric values in the column. Applicable data types include decimal, double, integer, and real. This function only includes the Good rows.</td></tr><tr><td>Column</td><td>Checksum</td><td>Checksum of the values in the column.</td></tr></table>	Data object	Audit function	Description	Table or output schema	Count	This function collects two statistics: • Good count for rows that were successfully processed. • Error count for rows that generated some type of error if you enabled error handling.	Column	Sum	Sum of the numeric values in the column. Applicable data types include decimal, double, integer, and real. This function only includes the Good rows.	Column	Average	Average of the numeric values in the column. Applicable data types include decimal, double, integer, and real. This function only includes the Good rows.	Column	Checksum	Checksum of the values in the column.
Data object	Audit function	Description														
Table or output schema	Count	This function collects two statistics: • Good count for rows that were successfully processed. • Error count for rows that generated some type of error if you enabled error handling.														
Column	Sum	Sum of the numeric values in the column. Applicable data types include decimal, double, integer, and real. This function only includes the Good rows.														
Column	Average	Average of the numeric values in the column. Applicable data types include decimal, double, integer, and real. This function only includes the Good rows.														
Column	Checksum	Checksum of the values in the column.														
Audit label	The unique name in the data flow that the software generates for the audit statistics collected for each audit function that you define. You use these labels to define audit rules for the data flow.															
Audit rule	A Boolean expression in which you use audit labels to verify the job. If you define multiple rules in a data flow, all rules must succeed or the audit fails.															
Actions on audit failure	One or more of three ways to generate notification of an audit rule (or rules) failure: email, custom script, raise exception.															

18.5.4.1.1 Audit function

This section describes the data types for the audit functions and the error count statistics.

Data types

The following table shows the default data type for each audit function and the permissible data types. You can change the data type in the [Properties](#) window for each audit function in the Designer.

Audit functions	Default data type	Allowed data types
Count	Integer	Integer

Audit functions	Default data type	Allowed data types
Sum	Type of audited column	Integer, Decimal, Double, Real
Average	Type of audited column	Integer, Decimal, Double, Real
Checksum	Varchar(128)	Varchar(128)

Error count statistic

When you enable a Count audit function, the software collects two types of statistics:

- *Good* row count for rows processed without any error.
- *Error* row count for rows that the job could not process but ignores those rows to continue processing. One way that error rows can result is when you specify the *Use overflow file* option in the Source Editor or Target Editor.

18.5.4.1.2 Audit label

The software generates a unique name for each audit function that you define on an audit point. You can edit the label names. You might want to edit a label name to create a shorter meaningful name or to remove dashes, which are allowed in column names but not in label names.

Generating label names

If the audit point is on a table or output schema, the software generates the following two labels for the audit function Count:

```
$Count<_objectname>
$CountError<_objectname>
```

If the audit point is on a column, the software generates an audit label with the following format:

```
$ <auditfunction_objectname>
```

If the audit point is in an embedded data flow, the labels have the following formats:

```
$Count<_objectname_embeddedDFname>
$CountError<_objectname_embeddedDFname>
$<auditfunction_objectname_embeddedDFname>
```

Editing label names

You can edit the audit label name when you create the audit function and before you create an audit rule that uses the label.

If you edit the label name after you use it in an audit rule, the audit rule does not automatically use the new name. You must redefine the rule with the new name.

18.5.4.1.3 Audit rule

An audit rule is a Boolean expression which consists of a Left-Hand-Side (LHS), a Boolean operator, and a Right-Hand-Side (RHS).

- The LHS can be a single audit label, multiple audit labels that form an expression with one or more mathematical operators, or a function with audit labels as parameters.
- The RHS can be a single audit label, multiple audit labels that form an expression with one or more mathematical operators, a function with audit labels as parameters, or a constant.

The following Boolean expressions are examples of audit rules:

```
$Count_CUSTOMER = $Count_CUSTDW  
$Sum_ORDER_US + $Sum_ORDER_EUROPE = $Sum_ORDER_DW  
round($Avg_ORDER_TOTAL) >= 10000
```

18.5.4.1.4 Audit notification

You can choose any combination of the following actions for notification of an audit failure. If you choose all three actions, the software executes them in this order:

Action	Description
Email to list	The software sends a notification of which audit rule failed to the email addresses that you list in this option. Use a comma to separate the list of email addresses. You can specify a variable for the email list. This option uses the smtp_ function to send email. Therefore, you must define the server and sender for the Simple Mail Transfer Protocol (SMTP) in the Server Manager.
Script	The software executes the custom script that you create in this option.

Action	Description
Raise exception	The job fails if an audit rule fails, and the error log shows which audit rule failed. The job stops at the first audit rule that fails. This action is the default. You can use this audit exception in a try/catch block. You can continue the job execution in a try/catch block. If you clear this action and an audit rule fails, the job completes successfully and the audit does not write messages to the job log. You can view which rule failed in the Auditing Details report in the Metadata Reporting tool.

Related Information

[Viewing audit results \[page 345\]](#)

18.5.4.2 Opening the Audit window

Open the [Audit](#) window from one of the following places in the Designer:

- From the Data Flows tab of the object library, right-click on a data flow name and select the [Auditing](#) option.
- In the workspace, right-click on a data flow icon and select the [Auditing](#) option.
- When a data flow is open in the workspace, click the [Audit](#) icon in the toolbar.

When you first open the [Audit](#) window, the Label tab displays the sources and targets in the data flow. If your data flow contains multiple consecutive query transforms, the [Audit](#) window shows the first query.

Click the icons on the upper left corner of the Label tab to change the display.

Icon	Tool tip	Description
	Collapse All	Collapses the expansion of the source, transform, and target objects.
	Show All Objects	Displays all the objects within the data flow.
	Show Source, Target and first-level Query	Default display which shows the source, target, and first-level query objects in the data flow. If the data flow contains multiple consecutive query transforms, only the first-level query displays.
	Show Labelled Objects	Displays the objects that have audit labels defined.

18.5.4.3 Defining audit points, rules, and action on failure

1. Access the [Audit](#) window.
2. Define audit points. On the Label tab, right-click on an object that you want to audit and choose an audit function or Properties.

When you define an audit point, the software generates the following:

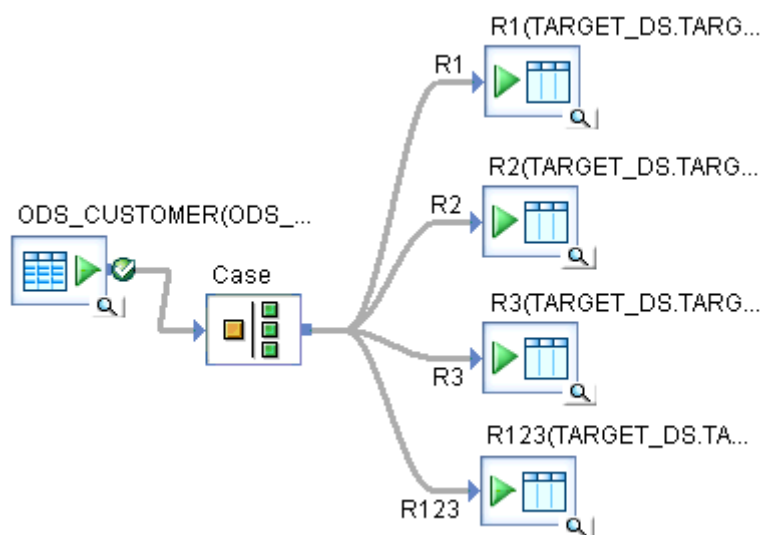
- An audit icon on the object in the data flow in the workspace
- An audit label that you use to define audit rules.

In addition to choosing an audit function, the Properties window allows you to edit the audit label and change the data type of the audit function.

For example, the data flow Case_DF has the following objects and you want to verify that all of the source rows are processed by the Case transform.

- Source table ODS_CUSTOMER
 - Four target tables:
 - R1 contains rows where ODS_CUSTOMER.REGION_ID = 1
 - R2 contains rows where ODS_CUSTOMER.REGION_ID = 2
 - R3 contains rows where ODS_CUSTOMER.REGION_ID = 3
 - R123 contains rows where ODS_CUSTOMER.REGION_ID IN (1, 2 or 3)
- a. Right-click on source table ODS_CUSTOMER and choose [Count](#).

The software creates the audit labels \$Count_ODS_CUSTOMER and \$CountError_ODS_CUSTOMER, and an audit icon appears on the source object in the workspace.



- b. Similarly, right-click on each of the target tables and choose [Count](#). The Audit window shows the following audit labels.

Target table	Audit function	Audit label
ODS_CUSTOMER	Count	\$Count_ODS_CUSTOMER
R1	Count	\$Count_ R1
R2	Count	\$Count_ R2
R3	Count	\$Count_ R3
R123	Count	\$Count_ R123

- c. If you want to remove an audit label, right-click on the label, and the audit function that you previously defined displays with a check mark in front of it. Click the function to remove the check mark and delete the associated audit label.

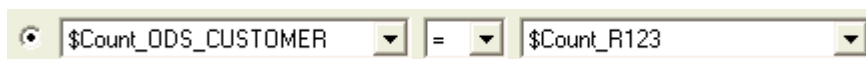
When you right-click on the label, you can also select Properties, and select the value (No Audit) in the [Audit function](#) drop-down list.

3. Define audit rules. On the Rule tab in the [Audit](#) window, click [Add](#) which activates the expression editor of the Auditing Rules section.

If you want to compare audit statistics for one object against one other object, use the expression editor, which consists of three text boxes with drop-down lists:

- a. Select the label of the first audit point in the first drop-down list.
- b. Choose a Boolean operator from the second drop-down list. The options in the editor provide common Boolean operators. If you require a Boolean operator that is not in this list, use the Custom expression box with its function and smart editors to type in the operator.
- c. Select the label for the second audit point from the third drop-down list. If you want to compare the first audit value to a constant instead of a second audit value, use the Customer expression box.

For example, to verify that the count of rows from the source table is equal to the rows in the target table, select audit labels and the Boolean operation in the expression editor as follows:



If you want to compare audit statistics for one or more objects against statistics for multiple other objects or a constant, select the Custom expression box.

- a. Click the ellipsis button to open the full-size smart editor window.
- b. Click the [Variables](#) tab on the left and expand the [Labels](#) node.
- c. Drag the first audit label of the object to the editor pane.
- d. Type a Boolean operator
- e. Drag the audit labels of the other objects to which you want to compare the audit statistics of the first object and place appropriate mathematical operators between them.
- f. Click [OK](#) to close the smart editor.
- g. The audit rule displays in the Custom editor. To update the rule in the top Auditing Rule box, click on the title "Auditing Rule" or on another option.
- h. Click [Close](#) in the Audit window.

For example, to verify that the count of rows from the source table is equal to the sum of rows in the first three target tables, drag the audit labels, type in the Boolean operation and plus signs in the smart editor as follows:

```
Count_ODS_CUSTOMER = $Count_R1 + $Count_R2 + $Count_R3
```

4. Define the action to take if the audit fails.

You can choose one or more of the following actions:

Action	Description
Raise exception	The job fails if an audit rule fails and the error log shows which audit rule failed. This action is the default. If you clear this option and an audit rule fails, the job completes successfully and the audit does not write messages to the job log. You can view which rule failed in the Auditing Details report in the Metadata Reporting tool.
Email to list	The software sends a notification of which audit rule failed to the email addresses that you list in this option. Use a comma to separate the list of email addresses. You can specify a variable for the email list.
Script	The software executes the script that you create in this option.

5. Execute the job.

The *Execution Properties* window has the *Enable auditing* option checked by default. Clear this box if you do not want to collect audit statistics for this specific job execution.

6. Look at the audit results.

You can view passed and failed audit rules in the metadata reports. If you turn on the audit trace on the Trace tab in the *Execution Properties* window, you can view all audit results on the Job Monitor Log.

Related Information

[Auditing objects in a data flow \[page 335\]](#)

[Viewing audit results \[page 345\]](#)

18.5.4.4 Guidelines to choose audit points

The following are guidelines to choose audit points:

- When you audit the output data of an object, the optimizer cannot push down operations after the audit point. Therefore, if the performance of a query that is pushed to the database server is more important than gathering audit statistics from the source, define the first audit point on the query or later in the data flow.

For example, suppose your data flow has a source, query, and target objects, and the query has a WHERE clause that is pushed to the database server that significantly reduces the amount of data that returns to the software. Define the first audit point on the query, rather than on the source, to obtain audit statistics on the query results.

- If a pushdown_sql function is after an audit point, the software cannot execute it.
- You can only audit a bulkload that uses the Oracle API method. For the other bulk loading methods, the number of rows loaded is not available to the software.
- Auditing is disabled when you run a job with the debugger.
- You cannot audit NRDM schemas or real-time jobs.
- You cannot audit within an ABAP data flow, but you can audit the output of an ABAP data flow.
- If you use the CHECKSUM audit function in a job that normally executes in parallel, the software disables the DOP for the whole data flow. The order of rows is important for the result of CHECKSUM, and DOP processes the rows in a different order than in the source.

18.5.4.5 Auditing embedded data flows

You can define audit labels and audit rules in an embedded data flow. This section describes the following considerations when you audit embedded data flows:

- Enabling auditing in an embedded data flow
- Audit points not visible outside of the embedded data flow

18.5.4.5.1 Enabling auditing in an embedded data flow

If you want to collect audit statistics on an embedded data flow when you execute the parent data flow, you must enable the audit label of the embedded data flow.

18.5.4.5.1.1 Enabling auditing in an embedded data flow

1. Open the parent data flow in the Designer workspace.
2. Click on the Audit icon in the toolbar to open the Audit window
3. On the Label tab, expand the objects to display any audit functions defined within the embedded data flow. If a data flow is embedded at the beginning or at the end of the parent data flow, an audit function might exist on the output port or on the input port.
4. Right-click the Audit function name and choose [Enable](#). You can also choose [Properties](#) to change the label name and enable the label.
5. You can also define audit rules with the enabled label.

18.5.4.5.2 Audit points not visible outside of the embedded data flow

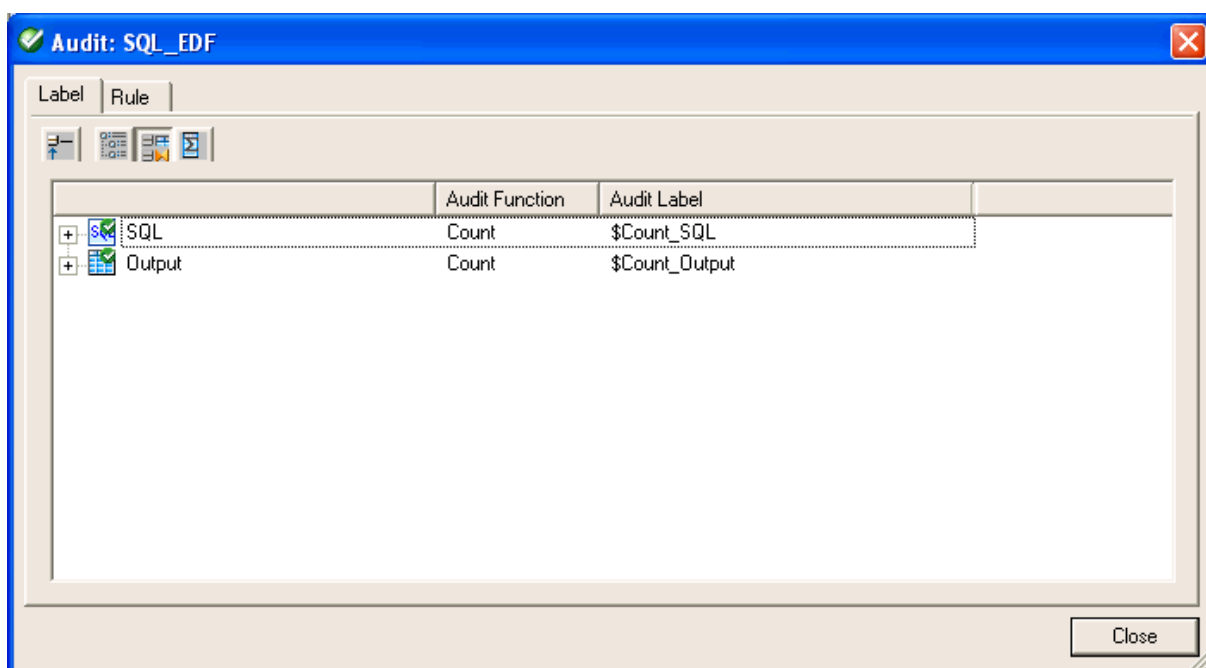
When you embed a data flow at the beginning of another data flow, data passes from the embedded data flow to the parent data flow through a single source. When you embed a data flow at the end of another data flow,

data passes into the embedded data flow from the parent through a single target. In either case, some of the objects are not visible in the parent data flow.

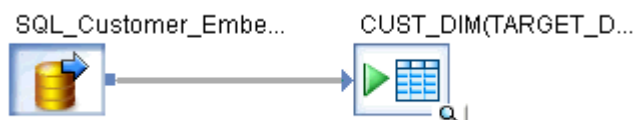
Because some of the objects are not visible in the parent data flow, the audit points on these objects are also not visible in the parent data flow. For example, the following embedded data flow has an audit function defined on the source SQL transform and an audit function defined on the target table.



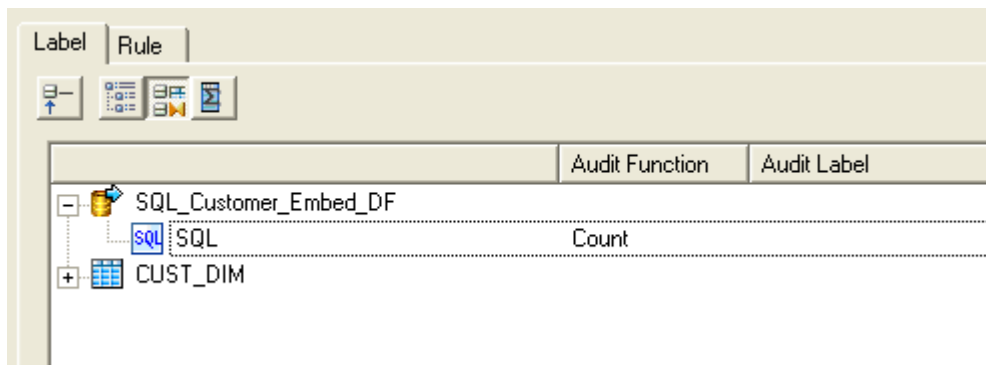
The following Audit window shows these two audit points.



When you embed this data flow, the target Output becomes a source for the parent data flow and the SQL transform is no longer visible.



An audit point still exists for the entire embedded data flow, but the label is no longer applicable. The following Audit window for the parent data flow shows the audit function defined in the embedded data flow, but does not show an Audit Label.



If you want to audit the embedded data flow, right-click on the audit function in the Audit window and select [Enable](#).

18.5.4.6 Resolving invalid audit labels

An audit label can become invalid in the following situations:

- If you delete the audit label in an embedded data flow that the parent data flow has enabled.
- If you delete or rename an object that had an audit point defined on it

18.5.4.6.1 Resolving invalid audit labels

1. Open the [Audit](#) window.
2. Expand the Invalid Labels node to display the individual labels.
3. Note any labels that you would like to define on any new objects in the data flow.
4. After you define a corresponding audit label on a new object, right-click on the invalid label and choose [Delete](#).
5. If you want to delete all of the invalid labels at once, right click on the Invalid Labels node and click on [Delete All](#).

18.5.4.7 Viewing audit results

You can see the audit status in one of the following places:

- Job Monitor Log
- If the audit rule fails, the places that display audit information depends on the [Action on failure](#) option that you chose:

Action on failure	Places where you can view audit information
Raise exception	Job Error Log, Metadata Reports
Email to list	Email message, Metadata Reports
Script	Wherever the custom script sends the audit messages, Metadata Reports

Related Information

[Job Monitor Log \[page 346\]](#)

[Job Error Log \[page 346\]](#)

[Metadata Reports \[page 347\]](#)

18.5.4.7.1 Job Monitor Log

If you set [Audit Trace](#) to **Yes** on the Trace tab in the Execution Properties window, audit messages appear in the Job Monitor Log. You can see messages for audit rules that passed and failed.

The following sample audit success messages appear in the Job Monitor Log when [Audit Trace](#) is set to **Yes**:

```
Audit Label $Count_R2 = 4. Data flow <Case_DF>.
Audit Label $CountError_R2 = 0. Data flow <Case_DF>.
Audit Label $Count_R3 = 3. Data flow <Case_DF>.
Audit Label $CountError_R3 = 0. Data flow <Case_DF>.
Audit Label $Count_R123 = 12. Data flow <Case_DF>.
Audit Label $CountError_R123 = 0. Data flow <Case_DF>.
Audit Label $Count_R1 = 5. Data flow <Case_DF>.
Audit Label $CountError_R1 = 0. Data flow <Case_DF>.
Audit Label $Count_ODS_CUSTOMER = 12. Data flow <Case_DF>.
Audit Label $CountError_ODS_CUSTOMER = 0. Data flow <Case_DF>.
Audit Rule passed ($Count_ODS_CUSTOMER = (($CountR1 + $CountR2 + $Count_R3)):
LHS=12, RHS=12. Data flow <Case_DF>.
Audit Rule passed ($Count_ODS_CUSTOMER = $CountR123): LHS=12, RHS=12. Data flow
<Case_DF>.
```

18.5.4.7.2 Job Error Log

When you choose the [Raise exception](#) option and an audit rule fails, the Job Error Log shows the rule that failed. The following sample message appears in the Job Error Log:

```
Audit rule failed <($Count_ODS_CUSTOMER = $CountR1)> for <Data flow Case_DF>.
```


18.5.4.7.3 Metadata Reports

You can look at the [Audit Status](#) column in the Data Flow Execution Statistics reports of the Metadata Report tool. This [Audit Status](#) column has the following values:

Value	Description
Not Audited	Statistics were not collected.
Passed	All audit rules succeeded. This value is a link to the Auditing Details report which shows the audit rules and values of the audit labels.
Information Collected	This status occurs when you define audit labels to collect statistics but do not define audit rules. This value is a link to the Auditing Details report which shows the values of the audit labels.
Failed	Audit rule failed. This value is a link to the Auditing Details report which shows the rule that failed and values of the audit labels.

Related Information

[Management Console Guide: Operational Dashboard Reports](#)

18.6 Data Quality

18.6.1 Overview of data quality

Data quality is a term that refers to the set of transforms that work together to improve the quality of your data by cleansing, enhancing, matching and consolidating data elements.

Data quality is primarily accomplished in the software using four transforms:

Transform	Description
Address Cleanse	Parses, standardizes, corrects, and enhances address data.
Data Cleanse	Parses, standardizes, corrects, and enhances customer and operational data.
Geocoding	Uses geographic coordinates, addresses, and point-of-interest (POI) data to append address, latitude and longitude, census, and other information to your records.
Match	Identifies duplicate records at multiple levels within a single pass for individuals, households, or corporations within multiple tables or databases and consolidates them into a single source.

Related Information

[Cleansing Package Builder \[page 348\]](#)

[Address Cleanse \[page 483\]](#)

[Geocoding \[page 378\]](#)

[Matching strategies \[page 401\]](#)

18.6.2 Data Cleanse

18.6.2.1 Cleansing Package Builder

The parsing rules and other information that define how to parse and standardize data are stored in a cleansing package.

The Cleansing Package Builder in SAP Information Steward provides a graphical user interface to create and refine cleansing packages. Create a cleansing package from scratch based on sample data or adapt an existing cleansing package or SAP-supplied cleansing package to meet your specific data cleansing requirements and standards.

i Note

The Data Cleansing Advisor feature uses the SAP-supplied Person_Firm cleansing package when processing data. Do not delete or rename the Person_Firm cleansing package. If you modify the SAP-supplied Person_Firm cleansing package, you may alter the results in Data Cleansing Advisor.

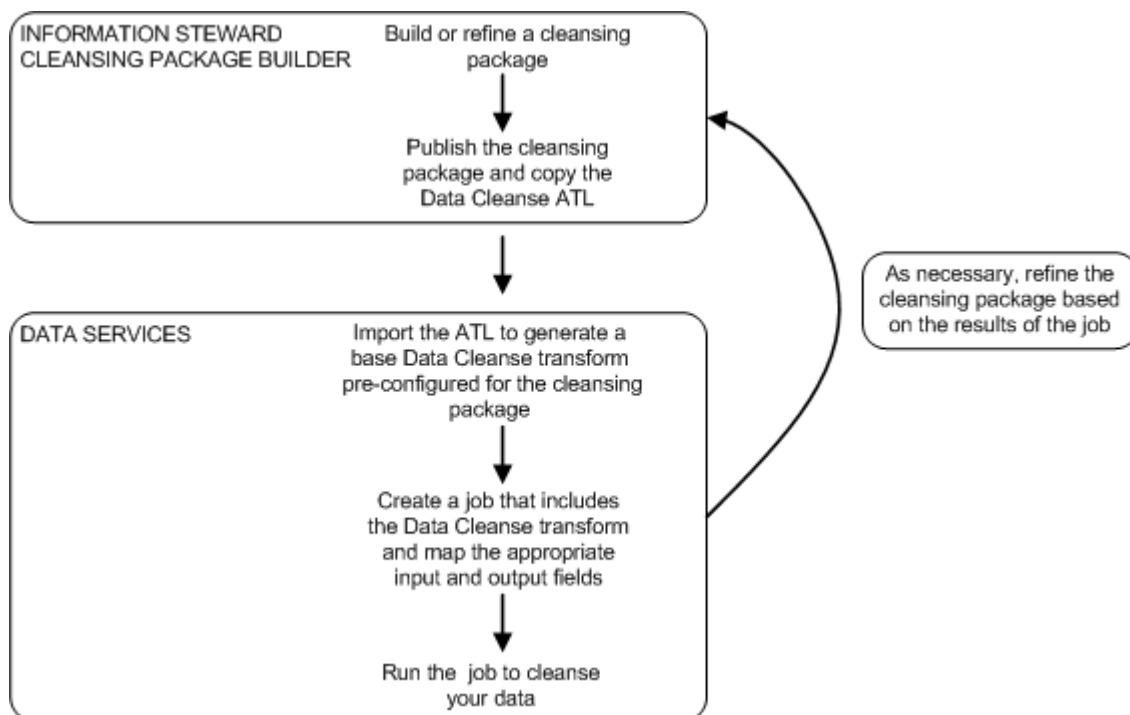
i Note

After creating one or more custom content types in Data Insight, the software creates a cleansing package named *CTID_CUSTOM_TYPE* in Cleansing Package Builder. Do not open or modify this cleansing package. Deleting the *CTID_CUSTOM_TYPE* cleansing package impacts the custom content types.

Cleansing Package Builder creates and publishes cleansing packages. The Data Cleanse transform in SAP Data Services then references the cleansing package for testing and production deployment.

Within a Data Services work flow, the Data Cleanse transform identifies and isolates specific parts of mixed data. It then parses and formats the data based on the referenced cleansing package settings and options you set in the transform.

The following diagram shows how Data Services and Information Steward work together to enable you to develop a cleansing package specific to your data requirements and then apply it when you cleanse your data.



18.6.2.2 Cleansing package lifecycle: develop, deploy and maintain

The process of developing, deploying, and maintaining a cleansing package is the result of action and communication between the Data Services administrator, Data Services tester, and Cleansing Package Builder data steward. The exact roles, responsibilities, and titles vary by organization, but often include the following:

Role	Responsibility
Cleansing Package Builder data steward	Uses Cleansing Package Builder and has domain knowledge to develop and refine a cleansing package for a specific data domain.
Data Services tester	Uses the Data Services transform in a Data Services test environment to cleanse data. Sends results to the Cleansing Package Builder data steward for verification and refinement.
Data Services administrator	Uses the Data Cleanse transform in a Data Services production environment to cleanse data based on the rules and standards defined in the selected cleansing package.

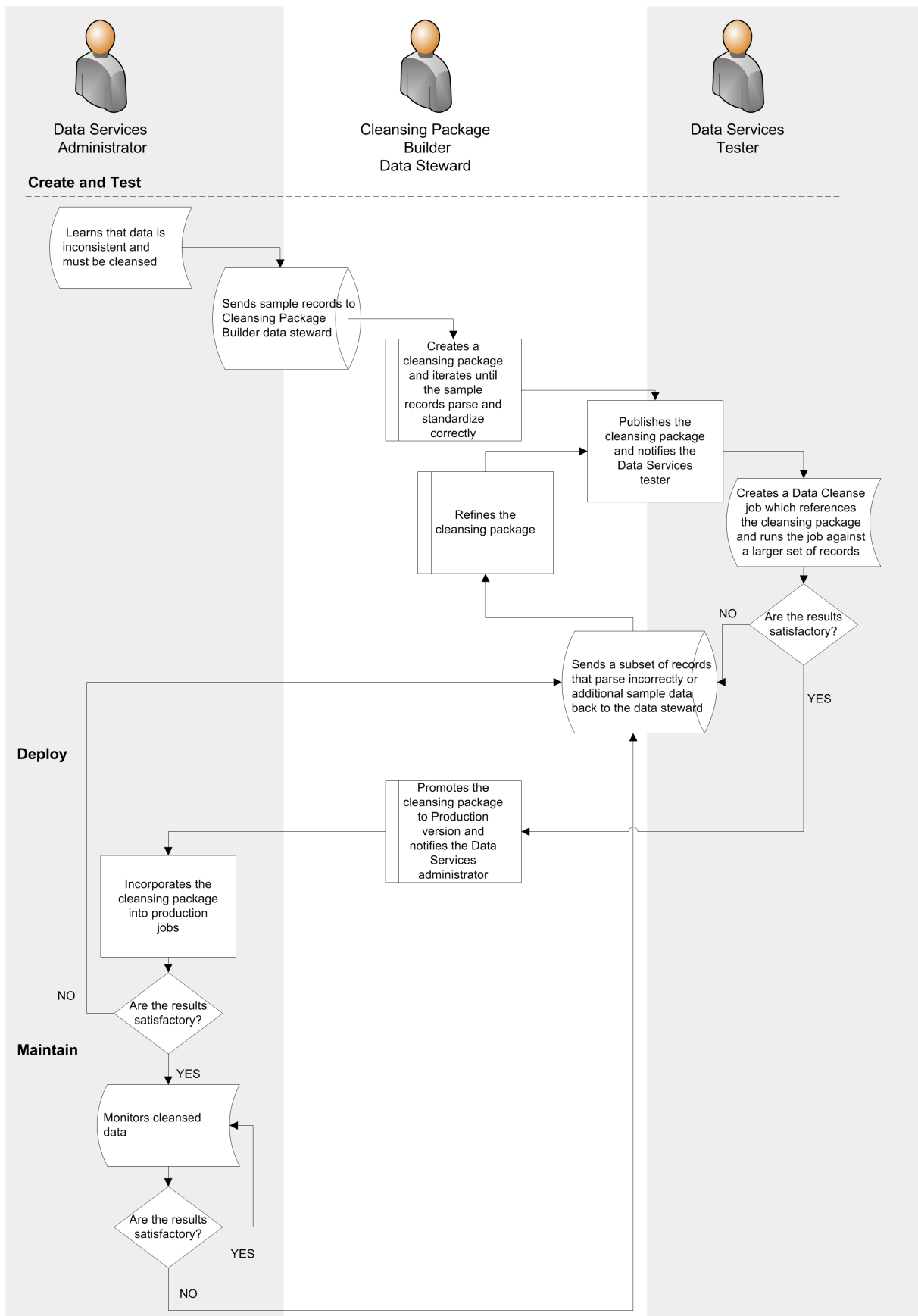
There are typically three iterative phases in a cleansing package workflow: Create and test (develop), deploy, and maintain.

Phases in cleansing package workflow

Phase	Custom cleansing package	Person and firm cleansing package
Create and test (develop)	The data steward creates a custom cleansing package using the Custom Cleansing Package Wizard. The data steward tests and refines the cleansing package in Cleansing Package Builder based on the data cleansing parsing and standardization results that appear in the bottom pane in <i>Design</i> mode. When the data steward is satisfied with the results, the data steward publishes the cleansing package and sends it to the Data Services tester to test the cleansing package on a larger data set. The data steward reviews the unsatisfactory results (if any) from the Data Cleanse testing, and makes changes to the cleansing package. This process continues until the cleansing package produces the expected results in Data Services.	The data steward creates a person and firm cleansing package using a copy of a published cleansing package, or by creating one manually (not recommended). The data steward tests and refines the cleansing package in Cleansing Package Builder based on the data cleansing parsing and standardization results that appear in the sample data in the <i>Validate</i> tab of Cleansing Package Builder. When the data steward is satisfied with the results, the data steward publishes the cleansing package and sends it to the Data Services tester to test the cleansing package on a larger data set. The data steward reviews the unsatisfactory results (if any) from the Data Cleanse testing, and makes changes to the cleansing package. This process continues until the cleansing package produces the expected results in Data Services. Note The <i>Validate</i> tab is only available for multiple-domain person and firm cleansing packages. For single-domain cleansing packages, when the data steward is satisfied with the results, the data steward publishes the cleansing package and sends it to the Data Services tester to test the cleansing package on a larger data set. The data steward reviews the unsatisfactory results (if any) and makes changes to the cleansing package. This process continues until the cleansing package produces the expected results in Data Services.

Phase	Custom cleansing package	Person and firm cleansing package
Deployment	When everyone is satisfied with the results of testing, the data steward deploys the cleansing package to production using the promotion management tool in the Central Management Console (CMC). For more information about promotion management, see “Lifecycle Management” in the <i>Information platform services Administrator Guide</i>.	When everyone is satisfied with the results of testing, the data steward deploys the cleansing package to production using the promotion management tool in the CMC. For more information about promotion management, see “Lifecycle Management” in the <i>Information platform services Administrator Guide</i>.
Maintenance	The Information Steward administrator moves the cleansing package into the maintenance phase and makes updates only when the results of regularly scheduled jobs fall out of range or when there is new data.	The Information Steward administrator moves the cleansing package into the maintenance phase and makes updates only when the results of regularly scheduled jobs fall out of range or when there is new data.

A typical workflow is shown in the diagram below:



For more information about the process of moving a cleansing package from development to production to maintenance, see the *Information Steward Administrator Guide*.

For more information about how to use the promotion management tool in the Central Management Console (CMC) to move cleansing packages (referred to as promoting “Objects” or “InfoObjects”), see the *Business Intelligence platform Administrator Guide*.

18.6.2.3 Configuring the Data Cleanse transform

Prerequisites for configuring the Data Cleanse transform include:

- Access to the necessary cleansing package.
- Access to the ATL file transferred from Cleansing Package Builder.
- Input field and attribute (output field) mapping information for user-defined pattern matching rules defined in the [Reference Data](#) tab of Cleansing Package Builder.

To configure the Data Cleanse transform:

1. Import the ATL file transferred from Cleansing Package Builder.

Importing the ATL file brings the required information and automatically sets the following options:

- Cleansing Package
- Filter Output Fields
- Input Word Breaker
- Parser Configuration

i Note

You can install and use SAP-supplied person and firm cleansing package without modifications directly in Data Services. To do so, skip step 1 and manually set any required options in the Data Cleanse transform.

2. In the input schema, select the input fields that you want to map and drag them to the appropriate fields in the [Input](#) tab.
 - Name and firm data can be mapped either to discrete fields or multiline fields.
 - Custom data must be mapped to multiline fields.
 - Phone, date, email, Social Security number, and user-defined pattern data can be mapped either to discrete fields or multiline fields. The corresponding parser must be enabled.
3. In the [Options](#) tab, select the appropriate option values to determine how Data Cleanse will process your data.

If you change an option value from its default value, a green triangle appears next to the option name to indicate that the value has been changed.

The ATL file that you imported in step 1 sets certain options based on information in the cleansing package.

4. In the [Output](#) tab, select the fields that you want to output from the transform. In Cleansing Package Builder, output fields are referred to as attributes.

Ensure that you map any attributes (output fields) defined in user-defined patterns in Cleansing Package Builder reference data.

Related Information

[Transform configurations \[page 304\]](#)
[Reference Guide: Transforms, Data Cleanse](#)

18.6.2.4 Rank and prioritizing parsing engines

Configure the Data Cleanse transform to use specific parsers and to specify the order in which the parsers run. Before you rank and prioritize parsing engines, careful planning may be beneficial. For example, turning off parsers that you do not need significantly improves parsing speed and reduces the chances that the transform parses data incorrectly.

Change the parser order for a specific multiline input by modifying the corresponding parser sequence option in the Parser Configuration options group of the Data Cleanse transform. For example, to change the order of parsers for the *MULTILINE1* input field, modify the *Parser Sequence Multiline1* option. Use a pipe (|) delimiter to separate the values of the parsers.

To change the selected parsers or the parser order: select a parser sequence, click *OK* at the message and then use the *Ordered Options* dialog box to make your changes.

Related Information

[Ordered options editor \[page 306\]](#)

18.6.2.5 About parsing data

The Data Cleanse transform identifies and isolates a wide variety of data ranging from persons names to inventory items. In the Data Cleanse transform, map the input fields in your data to the appropriate input fields in the transform. Person and firm data, phone, date, email, and Social Security number data can be mapped to either discrete input fields or multiline input fields.

The following example shows how Data Cleanse parses product data from a multiline input field and displays it in discrete output fields. You can also configure the transform to display data in composite fields. For example, customize the composite field *Standard_Description* in Cleansing Package Builder to meet your needs.

Input data	Parsed data	
Glove ultra grip profit 2.3 large black synthetic leather elastic with Velcro Mechanix Wear	Product Category	Glove
	Size	Large
	Material	Synthetic Leather

Input data	Parsed data	
	Trademark	Pro-Fit 2.3 Series
	Cuff Style	Elastic Velcro
	Palm Type	Ultra-Grip
	Color	Black
	Vendor	Mechanix Wear
	Standard Description	Glove - Synthetic Leather, Black, size: Large, Cuff Style: Elastic Velcro, Ultra-Grip, Mechanix Wear

The following examples show how Data Cleanse parses name and firm data and displays it in discrete output fields. The data also can be displayed in composite fields which you can customize in Cleansing Package Builder to meet your needs.

Input data	Parsed data		Composite data
Ms Mary Ann Smith Jones, CPA Account Mgr. Jones Inc.	Prenome	Ms.	
	Given Name 1	Mary	Given_Name_Full
	Given Name 2	Anne	Mary Anne
	Family Name 1	Smith	Family_Name_Full
	Family Name 2	Jones	Smith Jones
	Honorary Postname	CPA	
	Title	Account Mgr.	
	Firm	Jones, Inc.	

Input data	Parsed data	
James Witt Jr 421-55-2424 jwitt@rdrindustries.com 507-555-3423 Aug 20, 2003	Given Name 1	James
	Family Name 1	Witt
	Maturity Postname	Jr.
	Social Security	421-55-2424
	Email Address	jwitt@rdrindustries.com
	Phone	507.555.3423
	Date	August 20, 2003

The Data Cleanse transform parses up to six names per record and two names per input field. For all six names the transform finds, it parses components such as prename, given names, family name, and postname. Then it sends the data to individual fields. The transform also parses:

- Up to six job titles per record
- Up to six firm names per record
- One firm name per input field

18.6.2.5.1 About parsing phone numbers

Data Cleanse parses both North American Numbering Plan (NANP) and international phone numbers.

Phone numbering systems differ around the world. When Data Cleanse parses a phone number, it outputs the individual components of the number into the appropriate output fields.

The most efficient parsing happens when the phone number has a valid country code that appears before the phone number. If the country code is not present, or it is not first in the string, Data Cleanse uses other resources to attempt to parse the phone number.

When a valid country code is not present, Data Cleanse parses phone numbers by first searching internationally. It uses ISO2 country codes, when available, along with the patterns defined in the cleansing package for each country to identify the country code. If the transform encounters a country that participates in the NANP, it automatically stops trying to parse the number as international. The transform then attempts to parse the phone number as a number from North America, comparing the phone number to commonly used patterns such as (234) 567-8901, 234-567-8901, and 2345678901.

Currently, the participating countries in the NANP include:

- AS - American Samoa
- AI - Anguilla
- AG - Antigua and Barbuda
- BS - The Bahamas
- BB - Barbados
- BM - Bermuda
- VG - British Virgin Islands
- CA - Canada
- KY - Cayman Islands
- DM - Dominica
- DO - Dominican Republic
- GD - Grenada
- GU - Guam
- JM - Jamaica
- MS - Montserrat
- MP - Northern Mariana Islands
- PR - Puerto Rico
- KN - Saint Kitts and Nevis
- LC - Saint Lucia
- VC - Saint Vincent and the Grenadines
- SX - Saint Maarten
- TT - Trinidad and Tobago
- TC - Turks and Caicos Islands
- US - United States
- VI - United States Virgin Islands

Data Cleanse gives you the option for formatting North American numbers on output, such as your choice of delimiters. However, Data Cleanse outputs international numbers as they are input, without any formatting. Also, Data Cleanse does not cross-compare to the address to see whether the country and city codes in the phone number match the address.

Related Information

[About one-to-one mapping \[page 365\]](#)

18.6.2.5.1.1 How Data Cleanse parses phone numbers

Set up international phone parsing in the Data Cleanse transform at the record level, job file level, and global level.

Data Cleanse attempts to parse phone data in a specific order using levels: Record, job file, and global. If that fails, it parses the phone data as a North American phone number.

During phone parsing, Data Cleanse attempts to parse any number that contains a North American Numbering Plan (NANP) country code as a North American phone number. It skips any remaining levels.

i Note

Ensure that you specify the country in the cleansing package. If you do not specify the country, Data Cleanse does not attempt to parse the phone numbers from the country.

During phone parsing, Data Cleanse attempts to parse any number that contains a North American Numbering Plan (NANP) country code as a North American phone number. It skips any remaining levels.

Level	Process
Record	Attempts to parse the phone number based on the ISO2 country code. Parsing at the record level is optional. If your data does not contain the ISO2 country code, perform the following steps: 1. Create a data flow and position a Global Address Cleanse transform before the Data Cleanse transform. 2. Set the Global Address Cleanse transform to assign the ISO2 country code to each record and output the code to the <i>ISO_Country_Code_2Char</i> output field. 3. Open the Data Cleanse transform and map the <i>ISO_Country_Code_2Char</i> to the dynamic input field, <i>Option_Country</i>, in your input mapping. Advantages of record level parsing Because Global Address Cleanse uses the record address to determine the ISO2 code, there is a good chance that the phone number is from the same country as the address. If Data Cleanse parses a phone number at the record level, it continues parsing phone data from the next record. If the transform is not able to parse phone data at the record level, it moves to the job file level to attempt to make a match and parse the phone data. If the transform recognizes that the country code is from the NANP, it does the following: • Stops international parsing • Skips the job file and global levels • Attempts to parse the phone data as a North American phone number

Level	Process
Job file	Attempts to parse phone data using a sequence of countries in the <i>ISO2 Country Code Sequence</i><i>ISO2_COUNTRY_CODE_SEQUENCE</i> option in the Phone Options group. Phone parsing at the job file level is optional. Set up a country sequence to specify the order in which Data Cleanse searches for phone data. • Data Cleanse attempts to parse phone data in the first country in the sequence. • If the transform does not find a match in the first country, it moves to the next country in the sequence. It continues attempting to make a match until it finds a matching country. <i>Global</i> is the default setting for the sequence. Ensure that you set <i>Global</i> as the last entry in the country sequence. Or, you can delete <i>Global</i> from the sequence. Advantages of job file level parsing If you know that your data is predominately from specific countries, you set your sequence using those countries only, and remove <i>Global</i>. Paring down the list of countries to the countries in your data eliminates the need for the transform to search all countries for a match. If the country code is from the NANP, Data Cleanse stops the international parsing, skips the <i>Global</i> level, and attempts to parse the phone data as a North American phone number. The transform attempts to parse the phone data in the record as a North American phone number if: • You delete <i>Global</i> from the sequence • Data Cleanse did not find a match to the countries in the sequence To skip this level of phone parsing, leave the <i>ISO2 Country Code Sequence</i><i>ISO2_COUNTRY_CODE_SEQUENCE</i> option blank. Data Cleanse then attempts to parse the phone data at the global level.

Level	Process
Global	Attempts to parse phone data using <i>Global</i> as the country. Parsing phone data at the global level is optional. There are two ways to set up for the global level: • Use only <i>Global</i> in the <i>ISO2 Country Code Sequence</i><i>ISO2_COUNTRY_CODE_SEQUENCE</i>. • Delete <i>Global</i> from the first <i>ISO2 Country Code Sequence</i><i>ISO2_COUNTRY_CODE_SEQUENCE</i> and then include <i>Global</i> in the next sequence. Advantage of global level parsing The global level helps when you are unsure of the countries represented in your data. Global is also helpful if your data does not have the ISO2 country codes. At the global level, the transform searches all countries that are listed in the cleansing package, except the NANP countries. i Note The global level does not include country codes as search criteria. If the country code is from the NANP, Data Cleanse stops the global parsing and attempts to parse the phone data as a North American phone number. If the transform still cannot parse the phone data after the global level, it attempts to parse the phone data as a North American phone number.

Related Information

[Configuring the Data Cleanse transform \[page 353\]](#)

[Ordered options editor \[page 306\]](#)

18.6.2.5.2 About parsing dates

Data Cleanse recognizes dates in a variety of date formats and parses dates into components.

Data Cleanse can parse up to six dates from your defined record. That is, Data Cleanse identifies up to six dates in the input, breaks those dates into components, and makes dates available as output in either the original format or a user-selected standard format.

Related Information

[About one-to-one mapping \[page 365\]](#)

18.6.2.5.3 About parsing Social Security numbers

Data Cleanse parses U.S. Social Security numbers (SSNs) that are either by themselves or on an input line surrounded by other text.

Data Cleanse outputs the individual components of a parsed Social Security number: The entire SSN, the area, the group, and the serial.

Data Cleanse parses Social Security numbers in two steps:

- 1. Identifies a potential SSN by looking for patterns.
- 2. Performs a validity check on the first five digits only.

Pattern	Digits per grouping	Delimited by
nnnnnnnnnn	9 consecutive digits	N/A
nnn nn nnnn	3, 2, and 4 (for area, group, and serial)	spaces
nnn-nn-nnnn	3, 2, and 4 (for area, group, and serial)	all supported delimiters

The following table contains descriptions of the validity test outcomes.

Outcome	Description
Pass	Data Cleanse successfully parses the data and outputs the Social Security number to an SSN output field.

Fail	Data Cleanse does not parse the data because it is not a valid Social Security number. It outputs the data as extra, unparsed data.

i Note

Data Cleanse uses the U.S. government definition of a Social Security number to determine validity.

Check validity

When performing a validity check, Data Cleanse does not verify that a particular 9-digit Social Security number has been issued, or that it is the correct number for any named person. Instead, it validates only the first five digits (area and group). Data Cleanse does not validate the last four digits (serial), except to confirm that they are digits.

Output valid SSNs

Data Cleanse outputs only Social Security numbers that pass its validation. If an SSN fails validation, Data Cleanse does not output the number. Instead, it passes the number as a parsed, but invalid, Social Security number.

SSA data

Data Cleanse validates the first five digits based on a table from the Social Security Administration (http://www.socialsecurity.gov/employer/ssns/HGJune2411_final.txt). The rules and data that guide this check are available at <http://www.ssa.gov/history/ssn/geocard.html>. The Social Security number information that Data Cleanse references is included in the cleansing package.

i Note

The Social Security administration no longer updates the table. The last time it was updated was in July 2011. Therefore, Social Security validation performed in Data Cleanse is based on data through July, 2011. For more information about the Social Security Administration number assignment process, see <http://www.ssa.gov/employer/randomizationfaqs.html>.

Related Information

[Reference Guide: Transforms](#), [Data Quality transforms](#), [Data Cleanse](#), [output fields](#)

18.6.2.5.4 About parsing e-mail addresses

Data Cleanse parses e-mail addresses that are input in discrete fields or combined with other data in a multiline field.

After Data Cleanse recognizes input data as an e-mail address, it parses the data and places the components into specific fields for output.

A simple example of an e-mail is `joex@sap.com`

Data Cleanse identifies the following data components in e-mail data, when present:

- User name
- Domain name
- Top domain and up to four more domains
- Host name

Data Cleanse looks at the relationships of the data components to parse e-mail data.

What Data Cleanse does

Data Cleanse can take the following actions:

- Parse an e-mail address located either in a discrete field or combined with other data in a multiline field.
- Break down the domain name into subelements.
- Verify that an e-mail address is properly formatted.
- Flag the address as belonging to an internet service provider (ISP).

What Data Cleanse does not verify

Data Cleanse does not verify certain aspects of an e-mail address based on the following criteria:

- Whether the domain name (the portion to the right of the @ sign) is registered.
- Whether an e-mail server is active at that address.
- Whether the user name (the portion to the left of the @ sign) is registered on that e-mail server (if any).
- Whether the personal name in the record can be reached at this e-mail address.

E-mail components

The output field where Data Cleanse places the data depends on the position of the data in the record. Data Cleanse follows the Domain Name System (DNS) in determining the correct output field.

❖ Example

An input record contains the following data: expat@london.home.office.city.co.uk. During processing, Data Cleanse outputs the data as described in the following table.

Output field	Output value
Email	expat@london.home.office.city.co.uk
Email_User	expat
Email_Domain_All	london.home.office.city.co.uk
Email_Domain_Top	uk
Email_Domain_Second	co
Email_Domain_Third	city
Email_Domain_Fourth	office
Email_Domain_Fifth	home
Email_Domain_Host	london

Related Information

[About one-to-one mapping \[page 365\]](#)

18.6.2.5.5 About parsing user-defined patterns

Data Cleanse parses patterns found in data such as part numbers, invoice numbers, and driver licenses.

Data Cleanse can parse any alphanumeric sequence for which you define a pattern.

The user-defined pattern matching (UDPM) parser looks for the pattern across each entire field.

Patterns are defined using regular expressions in the [Reference Data](#) tab of Cleansing Package Builder. Check with the cleansing package owner to determine any required mappings for input fields and output fields (attributes).

18.6.2.5.6 About parsing street addresses

Because Data Cleanse does not identify and parse address components, process your data with either Global Address Cleanse or USA Regulatory Address Cleanse transforms before input to Data Cleanse.

If you input address data to Data Cleanse, it usually outputs the data to Extra fields.

Related Information

[How address cleanse works \[page 484\]](#)

18.6.2.5.7 About parsing firm names

Data Cleanse parses firm names when firm names are alone in a field, or together with other data.

An exception to how Data Cleanse recombines contiguous word pieces is for words that end with an apostrophe and an S.

❖ Example

The transform breaks the input string Macy's into three tokens:

- MACY
- '
- S

The transform first combines the tokens and looks up the term MACY'S. If it cannot find the term, it looks up the term without the apostrophe: MACYS. If that search is not successful, the transform automatically keeps the tokens together: MACY'S. It then adds the FIRM_MISCELLANEOUS classification to the term.

Since Data Cleanse automatically keeps words ending with S together, it is not necessary to add all possessive firm names to the dictionary.

18.6.2.5.8 About parsing name and title data

Data Cleanse parses name and title data from input.

A name can consist of the following parts:

- Prenom
- Given name
- Family name
- Postname
- Other name parts

Data Cleanse accepts up to two names and titles as discrete components. Data Cleanse also accepts name and title data together with other data or alone in a field. The name line or multiline field may contain one or two names per field.

18.6.2.5.9 About one-to-one mapping

Data Cleanse uses one-to-one mapping where it outputs data parsed from certain discrete input fields to their corresponding output fields.

One-to-one mapping is an option in the Data Cleanse transform that controls how several parsers output the data.

The *One-to-one mapping* option is available for these parsers:

- Date
- Email
- Phone

When the option is set to *Yes*, the Data Cleanse transform outputs the data, parsed from certain discrete input fields, to their corresponding output fields. The output fields are "reserved" for parses from certain discrete input fields. When you enable the *One-to-one mapping* option, the software more clearly shows the mapping of the input field that held the original data based on the parsed data in the output field.

❖ Example

If the input data in *Phone1-5* are blank, and the *Phone6* field contains data, the transform outputs *Phone1-5* as blank and outputs the parsed data to *Phone6*.

When you enable *One-to-one mapping*, then all parsers that use this option are set to *Yes*. For example, you cannot set the *One-to-one mapping* option only for the phone parser.

When you set *One-to-one mapping* to *No*, the Data Cleanse transform parses and outputs the data in the order the data entered the parser. The data is not necessarily output to the same field that it was mapped to on output. The data is output to the first available field in the category.

Note

The following examples show date fields. The same behavior applies to phone and e-mail fields.

Example

Field	Input data	Output data when option is <i>No</i>	Output data when option is <i>Yes</i>
Date1	<blank>	1968/01/01	<blank>
Date2	1968/01/01	1968/02/02	1968/01/01
Date3	1968/02/02	1968/03/03	1968/02/02
Date4	<blank>	1968/04/04	<blank>
Date5	1968/03/03	<blank>	1968/03/03
Date6	1968/04/04	<blank>	1968/04/04

Multiline fields

The transform parses the discrete date, e-mail, and phone fields before the multiline fields, so that any unreserved fields can contain data from the multiline fields.

Example

Field	Input data	Output data when option is <i>No</i>	Output data when option is <i>Yes</i>
Date1	<blank>	1968/01/01	<blank>
Date2	1968/01/01	1968/02/02	1968/01/01
Date3	<blank>	1968/03/03	<blank>
Date4	1968/02/02	1968/04/04	1968/02/02
Date5	<blank>	<blank>	1968/03/03 (not reserved, so Multiline input can be added here)
Date6	<blank>	<blank>	1968/04/04 (not reserved, so Multiline input can be added here)
Multiline1	1968/03/03 1968/04/04	<blank>	<blank>

Extra fields

When the *One-to-one mapping* option is set to *Yes* and the input field contains multiple sets of data, only the first set of data is placed in the corresponding output field. All other sets of data are put in the Extra field.

❖ Example

Field	Input data	Output data when option is <i>No</i>	Output data when option is <i>Yes</i>
Date1	1968/01/01 1968/02/02 1968/03/03 1968/04/04 1968/05/05 1968/06/06 1968/07/07 1968/08/08	1968/01/01	1968/01/01
Date2	<blank>	1968/02/02	<blank>
Date3	<blank>	1968/03/03	<blank>
Date4	<blank>	1968/04/04	<blank>
Date5	<blank>	1968/05/05	<blank>
Date6	<blank>	1968/06/06	<blank>
Extra	<blank>	1968/07/07 1968/08/08	1968/02/02 1968/03/03 1968/04/04 1968/05/05 1968/06/06 1968/07/07 1968/08/08

Related Information

[About parsing dates \[page 360\]](#)

[About parsing e-mail addresses \[page 362\]](#)

[About parsing phone numbers \[page 356\]](#)

18.6.2.6 About standardizing data

Data Cleanse standardizes input data so that it has a consistent format.

Data Cleanse uses Cleansing Package Builder to define standard formats for individual variations of data. Additionally, Data Cleanse standardizes data to make the format more consistent.

Data Cleanse standardizes data using the following data characteristics: Case, punctuation, and abbreviations.

18.6.2.7 About assigning gender descriptions and prenames

Each variation in a cleansing package has a gender associated with it.

Data Cleanse assigns a gender description to each name and populates the generated output field *Gender*.

By default, the gender is unassigned. You can assign a gender to a variation in the Advanced mode of Cleansing Package Builder. Variations in the SAP-supplied name and firm cleansing package have been assigned genders.

The intelligence behind gender assignment lies partly in the application and partly in the cleansing package.

Data Cleanse gender descriptions include the following:

- Strong male
- Strong female
- Weak male
- Weak female
- Ambiguous

The Prenom output field always includes prenames that are part of the name input data. Additionally, when the *Assign Prenames* option is set to *Yes*, Data Cleanse populates the *Prenom* output field when a strong male or strong female gender is assigned to a variation.

Dual names

For dual names, Data Cleanse offers four additional gender descriptions:

- Female multi-name
- Male multi-name
- Mixed multi-name
- Ambiguous multi-name

Data Cleanse generates these genders based on the assigned genders of the two names. The following table shows how Data Cleanse assigns multi-name genders.

Dual name	First name gender	Second name gender	Dual name assigned gender
Bob and Sue Jones	strong male	strong female	mixed multi-name
Bob and Tom Jones	strong male	strong male	male multi-name
Sue and Sara Jones	strong female	strong female	female multi-name
Bob and Pat Jones	strong male	ambiguous	ambiguous multi-name

Chinese and Japanese given names

Data Cleanse parses a give name by intelligently combining given name characters as opposed to including the given name as a variation in the cleansing package.

Data Cleanse generates the gender by combining the gender of the individual characters that make up the given name, using the information in the following table.

	Strong female	Strong male	Weak female	Weak male	Ambiguous
Strong female	strong female	ambiguous	strong female	ambiguous	strong female
Strong male	ambiguous	strong male	ambiguous	strong male	strong male
Weak female	strong female	ambiguous	weak female	ambiguous	weak female
Weak male	ambiguous	strong male	ambiguous	weak male	weak male
Ambiguous	strong female	strong male	weak female	weak male	ambiguous

18.6.2.8 Prepare records for matching

Process data with the Data Cleanse transform before you perform matching with the Match transform.

We recommend to pre process records with Data Cleanse before a matching operation so that your data is standardized consistently. Standardization increases the accuracy of your matches.

The Data Cleanse transform generates Match Standards or alternates for many name and firm fields as well as all custom output fields. For example, Data Cleanse can identify Patrick and Patricia as potential matches for the name Pat. Match standards can help you overcome two types of matching problems:

- Alternate spellings such as Catherine and Katherine
- Nicknames such as Pat and Patrick

The following example shows how Data Cleanse prepares records for matching.

❖ Example

Data Source 1

Input record	Cleansed record	
Intl Marketing, Inc.	Given Name 1	Pat
Pat Smith, Accounting Mgr.	Match Standards	Patrick, Patricia
	Given Name 2	
	Family Name 1	Smith
	Title	Accounting Mgr.
	Firm	Intl. Mktg. Inc.

Notice that Data Source 1 has Patrick and Patricia as Match Standards.

Data Source 2

Input record	Cleansed record	
Smith, Patricia R.	Given Name 1	Patricia
International Marketing, Incorp.	Match Standards	

Input record	Cleansed record	
	Given Name 2	R
	Family Name 1	Smith
	Title	
	Firm	Intl. Mktg, Inc.

Notice that for Data Source 2, there are no Match Standards for the multi-word firm name.

When a cleansing package does not include an alternate, the Match Standard output field for that term is empty. In the case of a multi-word output such as a firm name, when none of the variations in the firm name have an alternate, then the Match Standard output is empty. However, if at least one variation has an alternate associated with it, the cleansing package populates Match Standards with the variation alternate.

18.6.2.9 Region-specific data

18.6.2.9.1 About domains

A domain describes a specific type of data or content.

Domains enhance the ability of Data Cleanse to cleanse data appropriately according to the cultural standards of a region.

Within an SAP-supplied person and firm cleansing package, each supported locale is a domain.

The following table illustrates how name parsing may vary by culture:

Domain	Name	Parsed Output			
		Given_Name1	Given_Name2	Family_Name1	Family_Name2
Spanish	Juan C. Sánchez	Juan	C.	Sánchez	
Portuguese	João A. Lopes	João		A.	Lopes
French	Jean Christophe Rousseau	Jean Christophe		Rousseau	
German	Hans Joachim Müller	Hans	Joachim	Müller	
English (U.S. and Canada)	James Andrew Smith	James	Andrew	Smith	

Data Cleanse automatically assigns each variation to the Global domain. Data Cleanse may also assign a variation to one or more other domains. A variation may have a different meaning in each domain. In other words, the properties associated with a given variation, such as standard form and classification, may be specific to a domain.

❖ Example

The variation “AG” has different meanings in German and in U.S. English.

- In German, “AG” is an abbreviation for “Aktiengesellschaft” (a firm type) and is cleansed as "AG"
- In English “AG” is an abbreviation for “Agriculture” and is cleansed as "Ag."

You control how Data Cleanse cleanses your data by specifying the domain or domains you want Data Cleanse to apply and in what order.

i Note

Multiple domains are supported only in person and firm cleansing packages version 4.1 or higher.

Variations in custom cleansing packages as well as person and firm cleansing packages created before Information Steward 4.1 are assigned only to the Global domain.

Global domain

The Global domain is a special content domain that contains all variations and their associated properties. If a variation is not associated with domain-specific information, the Global domain serves as the default domain.

When you add a variation, the transform initially adds it to the Global domain. As necessary, you can add it to other domains to assign domain-specific information. Add a variation to a domain other than the Global domain only when the variation has properties such as gender, classification, and standard form. These properties differ between the Global and other domains.

If you delete a variation from the Global domain, the software also deletes the variation from all other associated domains.

Controlling domain-specific cleansing

Set the [Content Domain Sequence](#) to control how Data Cleanse parses your domain-specific data such as name, gender, and title.

❖ Example

Consider how gender is applied for the name “Jean” based on the information in the table.

Name	Domain	Gender
Jean	Global	AMBIGUOUS
Jean	French	STRONG_MALE
Jean	English (United States and Canada)	WEAK_FEMALE

When you do not want to favor any domain-specific properties, select only [Global](#).

❖ Example

For the name “Jean”, when you select only [Global](#), the transform assigns an ambiguous gender. With only Global, the transform does not consider the French or the English domain-specific information.

When you have data from a single-domain region, specify a domain followed by *Global*.

❖ Example

For the name “Jean”, when you specify *EN_US* followed by *Global* (*EN_US|GLOBAL*), the name Jean is assigned a weak female gender.

When you have data from a multi-domain region, select the preferred sequence of domains ending with Global. For example, Benelux (Belgium, Netherlands, Luxembourg) includes the Dutch, French, and German domains.

Depending on your cleansing preference, order the domains in the desired sequence.

❖ Example

If you favor the Dutch domain, specify domains in the following order: *NL|FR|DE|GLOBAL*. When the transform encounters a variation in your data that has different properties in the selected content domains, the transform processes data in the following way:

- Uses the Dutch properties first when they exist in the data
- If no Dutch properties, uses the French properties when they exist in the data.
- If no French properties, uses the German properties when they exist in the data.
- If none of the three domains have specific properties, then Data Cleanse uses the properties specified in the Global domain.

Another example of a multi-domain region is Switzerland. Switzerland includes German, French, and Italian domains. Depending on your cleansing preference, you can order the domains in the desired sequence. For example, if you favor the German domain you may select *DE|FR|IT|GLOBAL*.

Specifying the content domain sequence

You can set the content domain sequence in three different levels: Default, Intermediate, and Top. You use the Data Cleanse, Global Address Cleanse, and Query transforms to set each level.

Level	Option
Default	Use the Data Cleanse transform option <i>Content Domain Sequence</i>, which is a required option. The transform uses the value in <i>Content Domain Sequence</i> when there are errors or when you do not specify the intermediate and top levels.

Level	Option
Intermediate	Include any of the following input fields in your field mapping: Option_Country Option_Language Option_Region These input fields are populated and mapped from the Global Address Cleanse transform output fields: ISO_Country_Code_2Char, Language, and Region1, respectively. If there are errors, or the top-level option is not specified, the transform uses the intermediate setting. Using these input fields is optional.
Top	Include the following input field in your input field mapping: Option_Content_Domain_Sequence. An upstream Query transform, or a different preprocessing step, populates the Option_Content_Domain_Sequence field. Using this input field is optional.

❖ Example

During processing, Data Cleanse assigns the content domain sequence based on the top-level option, if the option is set and is valid.

Level	Option	Output	Notes
Top	Input field Option_Content_Domain_Sequence = <i>FR GLOBAL</i>	The content domain sequence is French. All data that isn't identified as French defaults to the Global domain.	The Top level overrides the Intermediate and Default level settings unless the Top-level option is invalid. For example, the Top level is invalid when a word is misspelled or the level is not set, for example.

Level	Option	Output	Notes
Intermediate	Input fields: <i>Option_Country</i> = <i>MX</i> <i>Option_Language</i> is not set <i>Option_Region</i> is not set	The content domain sequence is Spanish, ES_MX, which is local to Mexico. All of the data that isn't identified as Spanish defaults to the Global domain.	The Intermediate level overrides the Default level, unless the Intermediate level contains invalid data, or is not set. In most cases, the transform can identify the content domain sequence based on the <i>Option_Country</i> input field only. However, there are certain countries that can use <i>Option_Language</i> or <i>Option_Region</i> to help determine the content domain sequence.
Default	Data Cleanse transform option: <i>Content Domain Sequence</i> = <i>EN_US GLOBAL</i>	The content domain sequence is English and local to United States and Canada. All data that isn't identified as English defaults to the Global domain.	The <i>Content Domain Sequence</i> option is required. The transform uses the setting when there are errors in the Top-level and Intermediate-level options.

When the transform uses the Intermediate level, some countries can use additional input data to determine the content domain sequence. The following table shows an example of this behavior.

<i>Option_Country</i> input field	Additional input fields	Assigned content domain sequence	Notes
CA (Canada)	<i>Option_Region</i> = QC	FR GLOBAL	The transform always assigns the content domain sequence to FR GLOBAL when <i>Option_Region</i> is set to Quebec (QC). Acceptable data includes Quebec or QC in any type of casing. Any other valid region, such as BC for British Columbia, returns a content domain sequence of <i>EN_US GLOBAL</i> .
	<i>Option_Region</i> = BC*	EN_US GLOBAL	

<i>Option_Country</i> input field	Additional input fields	Assigned content domain sequence	Notes
<i>BE</i> (Belgium)	<i>Option_Language</i> = <i>FR</i>	<i>FR GLOBAL</i>	The transform always assigns the content domain sequence to <i>FR GLOBAL</i> when <i>Option_Language</i> = <i>FR</i> . Any other specified language, such as <i>NL</i> for Dutch or <i>EN_GB</i> for British English, or blank, returns a content domain sequence of <i>NL GLOBAL</i> .
	<i>Option_Language</i> = <i>NL</i> *	<i>NL GLOBAL</i>	
<i>CH</i> (Switzerland)	<i>Option_Language</i> = <i>FR</i>	<i>FR GLOBAL</i>	The transform always assigns the content domain sequence to <i>FR GLOBAL</i> or <i>IT GLOBAL</i> when <i>Option_Language</i> = <i>FR</i> or <i>IT</i> , respectively. Any other specified language, such as <i>RU</i> for Russian, or blank, returns a content domain sequence of <i>DE GLOBAL</i> .
	<i>Option_Language</i> = <i>IT</i>	<i>IT GLOBAL</i>	
	<i>Option_Language</i> = <i>RU</i> *	<i>DE GLOBAL</i>	

i Note

*The content domain sequence is assigned independently from the output format.

Related Information

[About output format \[page 375\]](#)

[Reference Guide: Transforms, Dynamic transform settings](#)

18.6.2.9.2 About output format

The Data Cleanse transform uses specific output fields based on the specified domain, and formats the data in those fields according to the regional standards.

Based on the set domain and regional standards, Data Cleanse may output name fields in the following ways.

- **Compound given name:**
 - Combines a compound given name and outputs it to the *Given_Name1* field.
 - Outputs the first name of a compound give name to the *Given_Name1* field and the second name to the *Given_Name2* field.
- **Compound family name:**
 - Combines a compound family name and outputs it to the *Family_Name1* field.

- Outputs the first family name of a compound family name to the *Family_Name1* field and the second family name to the *Family_Name2* field.
- **Composite *Person* output field:**
 - Outputs a given name followed by the family name in the composite *Person* field.
 - Outputs the family name followed by the given name in the composite *Person* field.

The Data Cleanse transform requires that you specify an output format, even when your data is truly global.

When you have data from a single-domain region, specify the domain. For example, for Germany select *DE*, for China select *ZH*. When you have data from a multi-domain region, select the preferred domain. Your data may be formatted differently depending on the domain you select.

❖ Example

Data from the Philippines may be output in English or Spanish output formats. As shown in the following table, Data Cleanse outputs name data to different fields depending on the selected output format.

Input name: Juan Carlos Sanchez Cruz

Output formats:

Output field	Output format	
	English	Spanish
<i>Given_Name1</i>	Juan	Juan
<i>Given_Name2</i>	Carlos	Carlos
<i>Family_Name1</i>	Sánchez Cruz	Sánchez
<i>Family_Name2</i>		Cruz
<i>Person</i>	Juan Carlos Sánchez Cruz	Juan Carlos Sánchez Cruz

❖ Example

For Benelux data, you may choose to output your data in Dutch, French, or German output formats. The Data Cleanse transform outputs name data in different fields based on the selected output format.

Input name: H.D. Budjhawan

Output formats:

Output field	Output format		
	Dutch	French	German
<i>Given_Name1</i>	H.D.	H. D.	H.
<i>Given_Name2</i>			D.
<i>Family_Name1</i>	Budjhawan	Budjhawan	Budjhawan
<i>Family_Name2</i>			
<i>Person</i>	H.D. Budjhawan	H. D. Budjhawan	H. D. Budjhawan

Modify the existing output format or add a new domain-specific output format by modifying the appropriate rules in your cleansing package.

Related Information

[Reference Guide: Transforms, Data Quality transforms, Data Cleanse, Cleansing Package options](#)
[Reference Guide: Transforms, Dynamic transform settings](#)

18.6.2.9.3 Customize prenames per country

When the input name does not include a prename, Data Cleanse generates the English prenames Mr. and Ms. To modify these terms, add a Query transform following the Data Cleanse transform and use the `search_replace` function to replace the terms with region-appropriate prenames.

18.6.2.9.4 Personal identification numbers

Data Cleanse can identify U.S. Social Security numbers and separate them into discrete components. If your data includes personal identification numbers other than U.S. Social Security numbers, you can create user-defined pattern rules to identify the numbers. User-defined pattern rules are part of the cleansing package and are defined in the [Edit Reference Data](#) tab of Cleansing Package Builder.

User-defined pattern rules are parsed in Data Cleanse with the [UDPM](#) parser. U.S. Social Security numbers are parsed in Data Cleanse with the [SSN](#) parser.

18.6.2.9.5 Text width in output fields

Japanese and latin characters can be encoded in a variety of widths.

Many Japanese characters are represented in both fullwidth and halfwidth forms. Latin characters can be encoded in either a proportional or fullwidth form. In either case, the fullwidth form requires more space than the halfwidth or proportional form.

To standardize your data, use the [Character Width Style](#) option to set the character width for all output fields to either fullwidth or halfwidth.

The [Normal Width](#) value reflects the normalized character width based on script type. Therefore, some output fields contain halfwidth characters and other fields contain fullwidth characters.

❖ Example

All fullwidth Latin characters are standardized to their halfwidth forms. All halfwidth katakana characters are standardized to their fullwidth forms.

[Normal Width](#) does not require special processing and, therefore, is the most efficient setting.

i Note

Because the output width is based on the normalized width for the character type, the output data may be larger than the input data. Increase the column width in the target table when the output data is larger.

For template tables, selecting [Use NVARCHAR for VARCHAR columns in supported databases](#) changes the VARCHAR column type to NVARCHAR and allows for increased data size.

Related Information

[Reference Guide: Locales and Multi-byte Functionality](#), [Multi-byte support](#), [Column Sizing](#)

18.6.3 Geocoding

An introduction to the Geocoder transform features.

The Geocoder transform uses geographic coordinates expressed as latitude and longitude, addresses, and point-of-interest (POI) data. With the Geocoder transform, you can append addresses, latitude and longitude, census data, and other information to your data.

Based on mapped input fields, the Geocoder transform has three modes of geocode processing:

- Address geocoding
- Reverse geocoding
- POI textual search

Typically, you use the Geocoder transform with the Global Address Cleanse or USA Regulatory Address Cleanse transform.

i Note

We are removing the GeoCensus functionality from the USA Regulatory Address Cleanse transform and including the features in the Geocoder transform. Ensure that you upgrade existing data flows that use GeoCensus features in the USA Regulatory Address Cleanse transform to the Geocoder transform. Find instructions for upgrading from GeoCensus to the Geocoder transform in the *Upgrade Guide*.

Related Information

[Reference Guide: Transforms](#), [Data Quality transforms](#), [Geocoder](#)

Reference Guide: Transforms, Data Quality transforms, Geocoder, Geocoder fields

[GeoCensus \(USA Regulatory Address Cleanse \) \[page 604\]](#)

18.6.3.1 Address geocoding

In address geocoding mode, the Geocoder transform assigns geographic data to the address.

Based on the completeness of the input address data, the Geocoder transform can return multiple levels of latitude and longitude data. Appending different levels of latitude and longitude information to your data may help your organization target certain population sizes and other regional geographical data.

Prepare records for geocoding

The Geocoder transform works best when it has standardized and corrected address data. To obtain the most accurate information, place an address cleanse transform before the Geocoder transform in the work flow.

18.6.3.1.1 Latitude and longitude levels

The Geocoder transform returns primary number, postcode centroid, and locality centroid latitude and longitude levels.

Primary number level

If your input data has a complete address that includes the primary number, the Geocoder transform returns latitude and longitude coordinates to the exact location. Data Cleanse bases the exact location on the directory type, either range-based or parcel-based, that you subscribe to.

Primary Range Interpolated (PRI) assignment level: In general, the Geocoder transform uses geocoding directories to calculate latitude and longitude values for a house. Geocoder interpolates between the beginning and ending point of a line segment, where the line segment represents a range of houses. The latitude and longitude values may be slightly offset from the exact location from where the house actually exists.

i Note

If you want an address-level assignment, ensure that you map the *Primary_Number* input field and that it is not blank.

Primary Range Exact (PRE) assignment level: The Geocoder transform also supports parcel directories. Parcel directories contain the most precise and accurate latitude and longitude values available for addresses, depending on the available country data. Parcel data is stored as points. Rather than getting you near the house, it takes you to the exact door.

Postcode Centroid level

If an address has a postcode, Geocoder provides coordinates in the appropriate postcode area. Geocoder the following postcode values for assignment levels:

- [Postcode_Full](#) (PF)
- [Postcode2_Partial](#) (P2P)
- [Postcode1](#) (P1)

The assignment levels are based on the available country data.

Locality Centroid level

If an address has a locality, you receive coordinates in the appropriate locality area. The Geocoder transform has the following locality assignment levels:

- [Locality1](#) (l1)
- [Locality2](#) (l2)
- [Locality3](#) (l3)
- [Locality4](#) (l4)

The assignment levels are based on the available country data.

18.6.3.1.2 Address geocoding field mapping

When you map input fields for address geocoding, consider that the more input data you provide the better the results.

The following tables specify the input and output fields that are required, optional, or those that cannot be mapped in the address geocoding mode.

Input field mapping

Input field category	Address geocoding mode
Address	At least one required
Address POI	Optional
Latitude/Longitude	n/a
Max Records	n/a
Search Filter	n/a

Output field mapping

Output field category	Address geocoding mode
Address	n/a
Address POI	n/a
Assignment Level	Optional
Census	Optional
Distance	n/a
Info Code	Optional
Latitude/Longitude	Optional
Population	Optional
Results	n/a
Side of Street	Optional
Status Code	Optional

18.6.3.1.3 Address geocoding scenario

In the following address geocoding scenario, use an address and a point of interest to assign latitude and longitude to your data.

Required input fields

To assign latitude and longitude, ensure that you map the [Country](#) input field. However, the more input data you can provide, the better your results.

Category	Input field name
Address	<i>Country</i> (required)
	<i>Locality1–4</i>
	<i>Postcode1–2</i>
	<i>Primary_Name1–4</i>
	<i>Primary_Number</i>
	<i>Primary_Postfix1</i>
	<i>Primary_Prefix1</i>
	<i>Primary_Type1–4</i>
	<i>Region1–2</i>

Optional input fields

Category	Input field name
Address POI	<i>POI_Name</i>
	<i>POI_Type</i>

Available output fields

All output fields are optional.

Category	Output field name
Assignment Level	<i>Assignment_Level</i>
	<i>Assignment_Level_Locality</i>
	<i>Assignment_Level_Postcode</i>

Category	Output field name
Census	<i>Census_Tract_Block</i> <i>Census_Tract_Block_Group</i> <i>Gov_County_Code</i> <i>Gov_Locality1_Code</i> <i>Gov_Region1_Code</i> <i>Metro_Stat_Area_Code</i> <i>Minor_Div_Code</i> <i>Stat_Area_Code</i>
Info Code	<i>Info_Code</i>
Latitude/Longitude	<i>Latitude</i> <i>Latitude_Locality</i> <i>Latitude_Postcode</i> <i>Latitude_Primary_Number</i> <i>Longitude</i> <i>Longitude_Locality</i> <i>Longitude_Postcode</i> <i>Longitude_Primary_Number</i>
Population	<i>Population_Class_Locality1</i>
Side of Street	<i>Side_Of_Primary_Address</i>
Status Code	<i>Status_Code</i>

❖ Example

You map input field that contain the following data:

Input field name	Input value
<i>Country</i>	US
<i>Postcode1</i>	54601
<i>Postcode2</i>	4023
<i>Primary_Name1</i>	Front
<i>Primary_Number</i>	332
<i>Primary_Type1</i>	St.

The mapped output fields contain the following results:

Output field name	Output value
<i>Assignment_Level</i>	PRE
<i>Latitude</i>	43.811616
<i>Longitude</i>	-91.256695

Related Information

[Understanding your output \[page 397\]](#)

[Reference Guide: Data Quality fields, Geocoder fields, Input fields](#)

[Reference Guide: Data Quality fields, Geocoder fields, Output fields](#)

18.6.3.2 Reverse geocoding

In reverse geocoding mode, the Geocoder transform identifies the closest address or point of interest based on an input reference location.

Based on the input-mapping, reverse geocoding can process in four modes as described in the following table.

Input and Output	Description
Address input and single record output	To determine a unique reference point, use the address and optional address POI fields. Output only the single closest record that matches the search filter. If you do not map the search filter fields, the Geocoder transform defaults to the nearest point of interest or address.
Address input and multiple record output	To determine a unique reference point, use the address and optional address POI fields. Output the reference point data to output fields. Output the multiple closest records that match the search filter fields to the <i>Result_List</i> output field. To set the number of multiple records, enter a value in the <i>Max Records</i> option. Or, if you map the <i>Option_Max_Records</i> input field, the transform uses the value to determine the number of multiple records.
Latitude/longitude input and single record output	Use the latitude and longitude fields as the reference point, and output only the single closest record that matches the optional search filter fields. If you do not map the search filter fields, the Geocoder transform defaults to the nearest point of interest or address.

Input and Output	Description
Latitude/longitude input and multiple record output	Use the latitude and longitude fields as the reference point, and output multiple closest records that match the search filter fields to the Result_List field. The transform determines the number of multiple records by the Option_Max_Records input field when it is mapped and populated or the Max Records option. If you do not map the search filter fields, the Geocoder transform defaults to the nearest point of interest or address.

Note

- Mapping the [Option_Radius](#) and [Option_Distance_Unit](#) input fields lets you define the distance from the specified reference point and identify an area in which matching records are located.
- To find one or more locations that can be points of interest, addresses, or both, set the [Search_Filter_Name](#) or [Search_Filter_Type](#) input field. Setting these fields limits the output matches to your search criteria.
- To return a non-POI address only, enter **ADDR** in the [Search_Filter_Type](#) input field.
- To return a POI only, enter the POI name or type in the [Search_Filter_Type](#) input field.
- If you don't set a search filter, the transform returns both addresses and points of interest.

18.6.3.2.1 Reverse geocoding field mapping

When you map input fields for reverse geocoding, consider that the more input data you provide the better the results.

The following tables specify the input and output fields that are required, optional, or cannot be mapped in the reverse geocoding mode.

Input field mapping

Input field category	Address input	Address input	Latitude/longitude input	Latitude/longitude input
	Single output	Multiple output	Single output	Multiple output
Address	At least one required	At least one required	n/a	n/a
Address POI	Optional	Optional	n/a	n/a
Latitude/Longitude	n/a	n/a	At least one required	At least one required
Max Records	n/a	At least one required	n/a	At least one required
Search Filter	At least one required	At least one required	Optional	Optional

Output field mapping

Output field category	Address input	Address input	Latitude/longitude input	Latitude/longitude input
	Single output	Multiple output	Single output	Multiple output
Address	Optional	n/a	Optional	n/a
Address POI	Optional	n/a	Optional	n/a
Assignment Level	Optional	Optional	Optional	n/a
Census	Optional	n/a	Optional	n/a
Distance	Optional	n/a	Optional	n/a
Info Code	Optional	Optional	Optional	Optional
Latitude/ Longitude	Optional	Optional	Optional	n/a
Population	Optional	n/a	Optional	n/a
Results	n/a	Optional	n/a	Optional
Side of Street	Optional	Optional	Optional	n/a

18.6.3.2.2 Reverse geocoding scenario 1

Use latitude and longitude to find one or more addresses or points of interest.

Example

The following example illustrates a scenario using latitude and longitude and a search filter to output a single point of interest closest to the input latitude and longitude.

Input: You map input fields that contain the following data:

Input field name	Input value
Latitude	43.811616
Longitude	-91.256695
Search_Filter_Name	BUSINESS OBJECTS

Output: The mapped output fields contain the following results:

Output field name	Output value
Address_Line	332 FRONT ST
Assignment_Level	PRE

Output field name	Output value
Country_Code	US
Distance	1.3452
Locality1	LA CROSSE
Postcode	54601-4023
Postcode1	54601
Postcode2	4023
Primary_Name1	FRONT
Primary_Number	332
Primary_Type1	ST
POI_Name	BUSINESS OBJECTS
POI_Type	5800
Region1	WI

Related Information

[Required input fields \[page 387\]](#)

[Available output fields \[page 388\]](#)

[Optional input fields \[page 388\]](#)

18.6.3.2.2.1 Required input fields

Scenario 1

For a single-record result, both Latitude and Longitude input fields must be mapped. For multiple-record results, the Latitude, Longitude, and Option_Max_Records input fields must all be mapped.

Category	Single-record results	Multiple-record results
	Input field name	Input field name
Latitude/Longitude	Latitude	Latitude
	Longitude	Longitude
Max Records	n/a	Option_Max_Records

18.6.3.2.2 Optional input fields

Scenario 1

Category	Single-record results	Multiple-record results
	Input field name	Input field name
Search Filter	Option Radius	Option_Radius
	Search_Filter_Name	Search_Filter_Name
	Search_Filter_Type	Search_Filter_Type

18.6.3.2.3 Available output fields

Scenario 1

All output fields are optional.

Category	Single-record results	Multiple-record results
	Output field name	Output field name
Address	Address_Line	n/a
	Country_Code	
	Locality1–4	
	Postcode	
	Postcode1–2	
	Primary_Name1–4	
	Primary_Number	
	Primary_Postfix1	
	Primary_Prefix1	
	Primary_Range_High	
	Primary_Range_Low	
	Primary_Type1–4	
Address POI	Region1–2	n/a
	POI_Name	
	POI_Type	

Category	Single-record results	Multiple-record results
	Output field name	Output field name
Assignment Level	Assignment_Level	n/a
	Assignment_Level_Locality	
	Assignment_Level_Postcode	
Census	Census_Tract_Block	n/a
	Census_Tract_Block_Group	
	Gov_County_Code	
	Gov_Locality1_Code	
	Gov_Region1_Code	
	Metro_Stat_Area_Code	
	Minor_Div_Code	
	Stat_Area_Code	
Distance	Distance	n/a
Info Code	Info_Code	Info_Code
Latitude/Longitude	Latitude	n/a
	Latitude_Locality	
	Latitude_Postcode	
	Latitude_Primary_Number	
	Longitude	
	Longitude_Locality	
	Longitude_Postcode	
Population	Population_Class_Locality1	n/a
Results	n/a	Result_List
		Result_List_Count
Side of Street	Side_Of_Primary_Address	n/a

18.6.3.2.3 Reverse geocoding scenario 2

Use an address or point of interest to find one or more closest addresses or points of interest.

In addition, the Geocoder transform outputs latitude and longitude information for both the input reference point and the matching output results.

Example

The following example illustrates a scenario using an address and a search filter to output a single point of interest closest to the input address. The request is to find the closest bank (POI type 6000) to the input location. The transform also outputs latitude and longitude information for the output result.

Input: You map input fields that contain the following data.

Input field name	Input value
Country	US
Postcode1	55601
Postcode2	4023
Primary_Name1	Front
Primary_Number	332
Primary_Type1	St.
POI_Name	BUSINESS OBJECTS
Search_Filter_Type	6000

Output: The mapped output fields contain the following results:

Output field name	Output value
Address_Line	201 MAIN ST
Assignment_Level	PRE
Country_Code	US
Distance	0.4180
Locality1	LA CROSSE
POI_Name	US BANK
POI_Type	6000
Postcode1	54601
Primary_Name1	MAIN
Primary_Number	201
Primary_Type1	ST
Region1	WI

Related Information

[Required input fields \[page 391\]](#)

[Optional input fields \[page 391\]](#)

[Available output fields \[page 392\]](#)

18.6.3.2.3.1 Required input fields

Scenario 2

For required input fields, at least one input field in each category must be mapped.

Category	Single-record results	Multiple-record results
	Input field name	Input field name
Address	Address_Line	Latitude
	Country	Longitude
	Locality1–4	
	Postcode	
	Postcode1–2	
	Primary_Name1–4	
	Primary_Number	
	Primary_Postfix1	
	Primary_Prefix1	
	Primary_Range_High	
	Primary_Range_Low	
	Primary_Type1–4	
	Region1–2	
Max Records	n/a	Option_Max_Records
Search Filter	Option_Radius	Option_Radius
	Search_Filter_Name	Search_Filter_Name
	Search_Filter_Type	Search_Filter_Type

18.6.3.2.3.2 Optional input fields

Scenario 2

Category	Single-record results	Multiple-record results
	Input field name	Input field name
Address POI	POI_Name	POI_Name
	POI_Type	POI_Type

18.6.3.2.3.3 Available output fields

Scenario 2

All output fields are optional.

For a single-record result, the output fields are the results for the spatial search.

For multiple-record results, the output fields in the Assignment Level and Latitude/Longitude categories are the results for the reference address assignment. Output fields in the Results category are the results for the reverse geocoding.

Category	Single-record results	Multiple-record results
	Output field name	Output field name
Address	Address_Line	n/a
	Country_Code	
	Locality1–4	
	Postcode	
	Postcode1–2	
	Primary_Name1–4	
	Primary_Number	
	Primary_Postfix1	
	Primary_Prefix1	
	Primary_Range_High	
	Primary_Range_Low	
	Primary_Type1–4	
	Region1–2	
Address POI	POI_Name	n/a
	POI_Type	
Assignment Level	Assignment_Level	Assignment_Level
	Assignment_Level_Locality	Assignment_Level_Locality
	Assignment_Level_Postcode	Assignment_Level_Postcode

Category	Single-record results	Multiple-record results
	Output field name	Output field name
Census	Census_Tract_Block	n/a
	Census_Tract_Block_Group	
	Gov_County_Code	
	Gov_Locality1_Code	
	Gov_Region1_Code	
	Metro_Stat_Area_Code	
	Minor_Div_Code	
	Stat_Area_Code	
Distance	Distance	n/a
Info Code	Info_Code	Info_Code
Latitude/Longitude	Latitude	Latitude
	Latitude_Locality	Latitude_Locality
	Latitude_Postcode	Latitude_Postcode
	Latitude_Primary_Number	Latitude_Primary_Number
	Longitude	Longitude
	Longitude_Locality	Longitude_Locality
	Longitude_Postcode	Longitude_Postcode
	Longitude_Primary_Number	Longitude_Primary_Number
Population	Population_Class_Locality1	n/a
Results	n/a	Result_List
		Result_List_Count
Side of Street	Side_Of_Primary_Address	Side_Of_Primary_Address

18.6.3.3 POI textual search

In the point of interest (POI) textual search mode, the Geocoder transform uses address fields and POI name or type fields as search criteria to match with points of interest.

The transform outputs the results to the [Result_List XML](#) output field.

The transform determines the number of multiple records by the [Option_Max_Records](#) input field if it is mapped and populated, or the [Max Records](#) option.

18.6.3.3.1 POI textual search field mapping

The following tables specify the input and output fields that are required, optional, or cannot be mapped in the POI textual search mode. The more input data you can provide, the better results you will obtain.

Input field mapping

In the POI textual search mode, at least one input field in the address POI category must be mapped, which is the key difference between the POI textual search and address geocoding modes.

Input field category	Address geocoding mode
Address	At least one required
Address POI	At least one required
Latitude/Longitude	n/a
Max Records	At least one required
Search Filter	n/a

Output field mapping

Output field category	Address geocoding mode
Address	n/a
Address POI	n/a
Assignment Level	n/a
Census	n/a
Distance	n/a
Info Code	Optional
Latitude/Longitude	n/a
Population	n/a
Results	Optional
Side of Street	n/a

18.6.3.3.2 POI textual search scenario

Scenario: Use an address and point-of-interest information to identify a list of potential point-of-interest matches.

Number of output results: Multiple records. The number of records is determined by the Option_Max_Records input field (if populated), or the Max Records option.

The following sections describe the required input fields and available output fields to obtain results for this scenario. We also provide an example with sample data.

Required input fields

For required input fields, at least one input field in each category must be mapped. The Country field must be mapped. The more input data you can provide, the better results you will obtain.

Category	Input field name
Address	Country (required)
	Locality1–4
	Postcode1–2
	Primary_Name1–4
	Primary_Number
	Primary_Postfix1
	Primary_Prefix1
	Primary_Type1–4
Address POI	Region1–2
Address POI	POI_Name
	POI_Type
Max Records	Option_Max_Records

Optional input fields

Not applicable.

Available output fields

All output fields are optional.

Category	Output field name
Info Code	Info_Code

Category	Output field name
Results	Result_List
	Result_List_Count

Example

The following example illustrates a scenario using POI textual search to identify a list of potential point-of-interest matches (all “BUSINESS OBJECTS” records on Front Street).

Input: You map input fields that contain the following data:

Input field name	Input value
Country	US
Postcode1	54601
Postcode2	4023
Primary_Name1	Front
Primary_Type1	St.
POI_Name	BUSINESS OBJECTS
Optional_Max_Records	10

Output: The mapped output fields contain the following results with one record:

Output field name	Output value
Result_List	Output as XML; example shown below
Result_List_Count	1

Result_List XML: The XML result for this example has one record.

```
<RESULT_LIST>
  <RECORD>
    <ADDRESS_LINE>332 FRONT ST</ADDRESS_LINE>
    <ASSIGNMENT_LEVEL>PRE</ASSIGNMENT_LEVEL>
    <COUNTRY_CODE>US</COUNTRY_CODE>
    <DISTANCE>0.3340</DISTANCE>
    <LATITUDE>43.811616</LATITUDE>
    <LOCALITY1>LA CROSSE</LOCALITY1>
    <LONGITUDE>-91.256695</LONGITUDE>
    <POI_NAME>BUSINESS OBJECTS</POI_NAME>
    <POI_TYPE>5800</POI_TYPE>
    <POSTCODE>56001-4023</POSTCODE>
    <POSTCODE1>56001</POSTCODE1>
    <POSTCODE2>4023</POSTCODE2>
    <PRIMARY_NAME1>FRONT</PRIMARY_NAME1>
    <PRIMARY_NUMBER>332</PRIMARY_NUMBER>
    <PRIMARY_TYPE1>ST</PRIMARY_TYPE1>
    <RANKING>1</RANKING>
    <REGION1>WI</REGION1>
  </RECORD>
</RESULT_LIST>
```

```
</RESULT_LIST>
```

18.6.3.4 Understanding your output

Information about Geocoder output data including latitude and longitude, assignment level, and multi record output.

Latitude and longitude

Latitude and longitude data is represented by decimal degrees such as 12.12345, for example. The decimals indicate the number of degrees from the equator. Latitude represents locations north or south of the equator. Longitude represents locations east or west of the equator.

- Latitude: 0-90 degrees north or south of the equator
 - Negative number indicates south of the equator
 - Positive number indicates north of the equator
- Longitude: 0-180 degrees east or west of Greenwich Meridian in London, England
 - Negative number indicates location within 180 degrees west of Greenwich
 - Positive number indicates location within 180 degrees east of Greenwich

Assignment level

You can understand the accuracy of the assignment based on the Assignment_Level output field.

Output field	Description
PRE	First level of assignment. The finest depth of assignment available to the exact location.
PRI	Second level of assignment. The finest depth of assignment available to the primary address range or house number.
P1 L1	Most general level of assignment. General assignment based on the postcode, P1, or the locality, L1. P1 or L1 is output depending on the value you choose in the Best Assignment Level option.

Multiple results

When your output consists of multiple records, the Result_List output field data is based the available data. The output format is an XML string.

Available data

Category	Output field name
Address	Address_Line
	Country_Code
	Locality1-4
	Postcode
	Postcode1-2
	Primary_Name1-4
	Primary_Number
	Primary_Postfix1
	Primary_Prefix1
	Primary_Range_High
	Primary_Range_Low
	Primary_Type1-4
Address POI	Region1-2
	POI_Name
	POI_Type
Assignment Level	Assignment_Level
Distance	Distance
Latitude/Longitude	Latitude
	Longitude
Ranking	Ranking

Related Information

[Reference Guide: Transforms, Data Quality transforms, Geocoder, Geocoder options](#)
[Standardize Geocoder output data \[page 399\]](#)

18.6.3.5 Standardize Geocoder output data

The output geocode data from Geocoder should be standardized using either Global Address Cleanse or USA Regulatory Address Cleanse transforms.

The address data output from the Geocoder transform should be standardized, because the vendor data that Geocoder uses to assign geocode data is not standardized. Before the geo output address data is useful for further processing, standardize the output data by following the Geocoder transform with the applicable address cleanse transform.

When you have multiple record output results, the output is an XML string that may be nested. Unnest the XML output data before sending it to a down-stream Global Address Cleanse or USA Regulatory Address Cleanse transform.

18.6.3.6 Working with other transforms

Typically, the Geocoder transform is used in conjunction with the Global Address Cleanse or USA Regulatory Address Cleanse transform.

Global Address Cleanse transform

For the Geocoder transform to provide the most accurate information, make sure the Global Address Cleanse transform output fields are mapped to Geocoder transform input fields as follows:

Global Address Cleanse output	Geocoder input field
ISO_Country_Code_2Char	Country
Locality1	Locality1
Postcode1	Postcode1
Postcode2	Postcode2
Primary_Name1	Primary_Name1
Primary_Number	Primary_Number
Primary_Postfix1	Primary_Postfix1
Primary_Prefix1	Primary_Prefix1
Primary_Type1	Primary_Type1
Region1_Symbol	Region1

It is also recommended that the Global Address Cleanse transform standardization options are set as follows:

Global Address Cleanse standardization option	Value
Assign Locality	Convert

Global Address Cleanse standardization option	Value
Directional Style	Short
Primary Type Style	Short

→ Tip

Postal address versus administrative address: The Global Address Cleanse transform uses postal addresses, while the Geocoder transform vendors, HERE (formerly NAVTEQ) and TOMTOM, use administrative addresses. Although a postal alignment is performed with postal addresses, there are some gaps. For example, for one Canadian address, the locality from the Global Address Cleanse transform might return TORONTO, but the Geocoder directory stores NORTH YORK. In this case, when the Locality1 field is mapped on input, it results in an Address Not Found error. As a workaround for this issue, you can remove the Locality1 input field mapping to get a better geocoding assignment.

18.6.4 Match

18.6.4.1 Match and consolidation using Data Services and Information Steward

When manual review is necessary, use SAP Data Services to automatically implement the overall matching and consolidation process. This process stages the match groups for manual review in SAP Information Steward.

To implement this process, consider the following factors:

- Do you stage only suspect match groups (groups that do not meet a match score or threshold that you define) for manual review? Or do you stage all the match results including unique records for match review and let data stewards manually move suspect records or unique records to the group they should belong? Note that the target group in this case could be a suspect group or “high confidence” group.
- Do you want to enable best record creation within Information Steward? Or do you want the Information Steward manual review to only modify suspect match groups as necessary and have Data Services handle the best record creation after the manual review is complete?
- If you want to enable best record creation within Information Steward, when do you perform the best record creation for high confidence match groups?
 - Manually creating a best record for all match groups in a large data set might not be practical.
 - If you have already consolidated high-confidence match groups within Data Services, then you have already modified the original match group to include the best record values, and you cannot stage these modified match groups for manual review in Information Steward.

Consider the following approaches for match review:

- Engage data stewards and domain experts in manual review of suspect match groups only. They can modify the suspect match groups if necessary and create the best record for these match groups.
 1. The Data Services job performs automated matching, identifies suspect match groups, and populates only suspect match groups in a staging database. In this scenario, “high confidence match groups” must be separated from “suspect match groups”. The best record can be created for the “high confidence match groups” and those match groups can be consolidated within Data Services.

This process can be accomplished by routing the “high confidence match groups” to another Match transform that is configured to create best records out of already matched groups of records.

2. Data stewards and domain experts review the “suspect match groups” in Information Steward and make corrections as appropriate (for example unmatch or move a record). They can manually create the best record for these match groups.
 - Engage data stewards and domain experts in manual review and correction of all match groups, but keep the best record creation and consolidation logic in Data Services.
 1. The Data Services job performs automated matching and populates all match results into the staging database. A match result table in the staging database contains suspect match groups, “high confidence” match groups, and unique records.
 2. Data stewards and domain experts review the suspect match groups in Information Steward and make corrections as appropriate. They can move the records between suspect match groups and “high confidence” match groups, or include a unique record in the appropriate match group.
 3. After the match review is finished, all match groups are processed in another Data Services job which creates the best record and consolidates all match groups.
- This process could be accomplished by a Match transform that is configured to create best records out of already matched groups of records.

i Note

You can find examples of best record creation using Information Steward or Data Services by downloading the Data Quality Management Match blueprints. You can find more information about blueprints at <https://wiki.scn.sap.com/wiki/display/EIM/SAP+Data+Services+Blueprints>

18.6.4.2 Matching strategies

Develop a match strategy before you use the Match transform.

To help develop your matching strategy, answer the following questions:

- What does my data consist of, for example, customer data, international data?
- What fields do I want to compare, for example, last name, firm?
- What are the relative strengths and weaknesses of the data in those fields?

→ Tip

If you cleanse your data before matching, you get better results. Also, data profiling can help you determine the strengths and weaknesses of your data.

- What end result do I want when the match job is complete? For example, do I want one record per family, per firm?

The following table contains a few match strategies to consider.

Strategy	Description
Simple match	Use when your matching business rules consist of a single match criteria for identifying relationships in consumer, business, or product data.

Strategy	Description
Consumer Householding	Use when your matching business rules consist of multiple levels of consumer relationships, such as residential matches, family matches, and individual matches.
Corporate Householding	Use when your matching business rules consist of multiple levels of corporate relationships, such as corporate matches, subsidiary matches, and contact matches.
Multinational consumer match	Use when your data consists of multiple countries and your matching business rules are different for different countries.
Identify a person multiple ways	Use when your matching business rules consist of multiple match criteria for identifying relationships, and you want to find the overlap between all of those definitions.

Related Information

18.6.4.3 Match components

When you prepare for matching, you work with three basic match components.

The basic components of matching are:

- Match sets
- Match levels
- Match criteria

Match sets

The Match transform represents a match set; each match set can have its own match criteria and business rules.

Match sets enable you to control how the Match transform matches certain records, segregates records, and matches on records independently. For example, set up match sets so you match U.S. records differently than international records.

A match set has the following purposes:

- To allow select data into a given set of match criteria for possible comparison. For example, exclude blank Social Security Numbers and international addresses.
- To allow for related match scenarios to be stacked to create a multi-level match set.
- To allow for multiple match sets to be considered for association in an Associate match set.

Match levels

A match level indicates the type of matching that the transform performs. For example, match on individual, family, resident, and firm. A match level refers to the broad category of matching instead of specific criteria.

You can have as many match levels as you want. However, the Match wizard restricts you to three levels during setup. You can add more levels later. You can define each match level in a match set in a way that is increasingly more strict. Multi-level matching feeds only the records that match from match level to match level (for example, resident, family, individual) for comparison.

Match level	Description
Family	The Match transform determines whether two people are members of the same family based on their record data. The transform compares the last name and the address data of two records. A match means that the two records represent members of the same family. The result of the match is one record per family.
Individual	The Match transform determines whether two records represent the same person based on their record data. The transform compares the first name, last name, and address data. A match means that the two records represent the same person. The result of the match is one record per individual.
Resident	The Match transform determine whether two records should be considered members of the same residence based on their record data. The transform compares the address data. A match means that the two records represent members of the same household. Contrast this match type with the family match type, which also compares last-name data. The result of the match is one record per residence.
Firm	The Match transform determines whether two records reflect the same firm. This match type involves comparisons of firm name and address data. A match means that the two records represent the same firm. The result of the match is one record per firm.
Firm-Individual	The Match transform determines whether two records are for the same person at the same firm based on their record data. The transform compares the first name, last name, firm name, and address data. A match means that the two records reflect the same person at the same firm. The result of the match is one record per individual per firm.

Match criteria

Match criteria refers to the fields on which to match. Use match criteria options to specify business rules for matching on each of these fields. The options enable you to control how close to exact the data has to be before the transform considers the records as matching.

For example, a match criteria may require: First names to be at least 85% similar, allow a first name initial to match a spelled out first name, and allow a first name to match a middle name.

The following list describes suggested fields to include in your match criteria for each match type:

- Family level match criteria may include family (last) name and address, or family (last) name and telephone number.
- Individual level match criteria may include full name and address, full name and SSN, or full name and E-mail address.
- Firm level match criteria may include firm name and address, firm name and Standard Industrial Classification (SIC) Code, or firm name and Data Universal Numbering System (DUNS) number.

18.6.4.4 Match Wizard

18.6.4.4.1 Match wizard

The Match wizard can quickly set up match data flows, without requiring you to manually create each individual transform it takes to complete the task.

What the Match wizard does

The Match wizard:

- Builds all the necessary transforms to perform the match strategy you choose.
- Applies default values to your match criteria based on the strategy you choose.
- Places the resulting transforms on the workspace, connected to the upstream transform you choose.
- Detects the appropriate upstream fields and maps to them automatically.

What the Match wizard does not do

The Match wizard provides you with a basic match setup that in some cases, will require customization to meet your business rules.

The Match wizard:

- Does not alter any data that flows through it. To correct non-standard entries or missing data, place one of the address cleansing transforms and a Data Cleanse transform upstream from the matching process.
- Does not connect the generated match transforms to any downstream transform, such as a Loader. You are responsible for connecting these transforms.
- Does not allow you to set rule-based or weighted scoring values for matching. The Match wizard incorporates a "best practices" standard that set these values for you. You may want to edit option values to conform to your business rules.

Related Information

[Combination method \[page 444\]](#)

18.6.4.4.2 Before you begin

Prepare a data flow for the Match wizard

To maximize its usefulness, be sure to include the following in your data flow before you launch the Match wizard:

- Include a Reader in your data flow. You may want to match on a particular input field that our data cleansing transforms do not handle.
- Include one of the address cleansing transforms and the Data Cleanse transform. The Match wizard works best if the data you're matching has already been cleansed and parsed into discrete fields upstream in the data flow.
- If you want to match on any address fields, be sure that you pass them through the Data Cleanse transform. Otherwise, they will not be available to the Match transform (and Match Wizard). This rule is also true if you have the Data Cleanse transform before an address cleanse transform.

18.6.4.4.3 Use the Match Wizard

18.6.4.4.3.1 Selecting match strategy

The Match wizard begins by prompting you to choose a match strategy, based on your business rule requirements. The path through the Match wizard depends on the strategy you select here. Use these descriptions to help you decide which strategy is best for you:

- **Simple match.** Use this strategy when your matching business rules consist of a single match criteria for identifying relationships in consumer, business, or product data.
- **Consumer Householding.** Use this strategy when your matching business rules consist of multiple levels of consumer relationships, such as residential matches, family matches, and individual matches.
- **Corporate Householding.** Use this strategy when your matching business rules consist of multiple levels of corporate relationships, such as corporate matches, subsidiary matches, and contact matches.
- **Multinational consumer match.** Use this match strategy when your data consists of multiple countries and your matching business rules are different for different countries.

i Note

The multinational consumer match strategy sets up a data flow that expects Latin1 data. If you want to use Unicode matching, you must edit your data flow after it has been created.

- **Identify a person multiple ways.** Use this strategy when your matching business rules consist of multiple match criteria for identifying relationships, and you want to find the overlap between all of those definitions.

Source statistics

If you want to generate source statistics for reports, make sure a field that houses the physical source value exists in all of the data sources.

To generate source statistics for your match reports, select the [Generate statistics for your sources](#) checkbox, and then select a field that contains your physical source value.

Related Information

[Unicode matching \[page 476\]](#)

18.6.4.4.3.2 Identifying matching criteria

Criteria represent the data that you want to use to help determine matches. In this window, you will define these criteria for each match set that you are using.

Match sets compare data to find similar records, working independently within each break group that you designate (later in the Match wizard). The records in one break group are not compared against those in any other break group.

To find the data that matches all the fields, use a single match set with multiple fields. To find the data that matches only in a specific combination of fields, use multiple match sets with two fields.

When working on student or snowbird data, an individual may use the same name but have multiple valid addresses.

Select a combination of fields that best shows which information overlaps, such as the family name and the SSN.

Data1	Data2	Data3	Data4
R. Carson	1239 Whistle Lane	Columbus, Ohio	555-23-4333
Robert T. Carson	52 Sunbird Suites	Tampa, Florida	555-23-4333

1. Enter the number of ways you have to identify an individual. This produces the corresponding number of match sets (transforms) in the data flow.
2. The default match set name appears in the [Name](#) field. Select a match set in the [Match sets](#) list, and enter a more descriptive name if necessary.
3. For each match set, choose the criteria you want to match on.
Later, you will assign fields from upstream transforms to these criteria.
4. Select the option you want to use for comparison in the Compare using column. The options vary depending on the criteria chosen. The compare options are:
 - [Field similarity](#)

- [Word similarity](#)
 - [Numeric difference](#)
 - [Numeric percent difference](#)
 - [Geo proximity](#)
5. Optional: If you choose to match on Custom, enter a name for the custom criteria in the Custom name column.
 6. Optional: If you choose to match on Custom, specify how close the data must be for that criteria in two records to be considered a match. The values that result determine how similar you expect the data to be during the comparison process for this criteria only. After selecting a strategy, you may change the values for any of the comparison rules options in order to meet your specific matching requirements. Select one of the following from the list in the Custom exactness column:
 - [Exact](#): Data in this criteria must be exactly the same; no variation in the data is allowed.
 - [Tight](#): Data in this criteria must have a high level of similarity; a small amount of variation in the data is allowed.
 - [Medium](#): Data in this criteria may have a medium level of similarity; a medium amount of variation in the data is allowed.
 - [Loose](#): Data in this criteria may have a lower level of similarity; a greater amount of variation in the data is allowed.

18.6.4.4.3.3 Defining match levels

Match levels allow matching processes to be defined at distinct levels that are logically related. Match levels refer to the broad category of matching not the specific rules of matching. For instance, a residence-level match would match on only address elements, a family-level match would match on only Last Name, and the individual-level match would match on First Name.

Multi-level matching can contain up to three levels within a single match set defined in a way that is increasingly more strict. Multi-level matching feeds only the records that match from match level to match level (that is, resident, family, individual) for comparison.

To define match levels:

1. Click the top level match, and enter a name for the level, if you don't want to keep the default name. The default criteria is already selected. If you do not want to use the default criteria, click to remove the check mark from the box.
The default criteria selection is a good place to start when choosing criteria. You can add criteria for each level to help make finer or more precise matches.
2. Select any additional criteria for this level.
3. If you want to use criteria other than those offered, click [Custom](#) and then select the desired criteria.
4. Continue until you have populated all the levels that you require.

18.6.4.4.3.4 Selecting countries

Select the countries whose postal standards may be required to effectively compare the incoming data. The left panel shows a list of all available countries. The right panel shows the countries you already selected.

1. Select the country name in the All Countries list.
2. Click [Add](#) to move it into the Selected Countries list.
3. Repeat steps 1 and 2 for each country that you want to include.

You can also select multiple countries and add them all by clicking the [Add](#) button.

The countries that you select are remembered for the next Match wizard session.

18.6.4.4.3.5 Grouping countries into tracks

Create tracks to group countries into logical combinations based on your business rules (for example Asia, Europe, South America). Each track creates up to six match sets (Match transforms).

1. Select the number of tracks that you want to create. The [Tracks](#) list reflects the number of tracks you choose and assigns a track number for each.
2. To create each track, select a track title, such as Track1.
3. Select the countries that you want in that track.
4. Click [Add](#) to move the selected countries to the selected track.

Use the COUNTRY UNKNOWN (__) listing for data where the country of origin has not been identified.

Use the COUNTRY OTHER (--) listing for data whose country of origin has been identified, but the country does not exist in the list of selected countries.

5. From Match engines, select one of the following engines for each track:

i Note

All match transforms generated for the track will use the selected Match engine.

- [LATINI](#) (Default)
- [CHINESE](#)
- [JAPANESE](#)
- [KOREAN](#)
- [TAIWANESE](#)
- [OTHER_NON_LATINI](#)

The [Next](#) button is only enabled when all tracks have an entry and all countries are assigned to a track.

18.6.4.4.3.6 Selecting criteria fields

Select and deselect criteria fields for each match set and match level you create in your data flow. These selections determine which fields are compared for each record. Some criteria may be selected by default, based on the data input.

If there is only one field of the appropriate content type, you will not be able to change the field for that criteria within the Match Wizard.

To enable the [Next](#) button, you must select at least one non-match-standard field.

1. For each of the criteria fields you want to include, select an available field from the drop-down list, which contains fields from upstream source(s). The available fields are limited to the appropriate content types for that criteria. If no fields of the appropriate type are available, all upstream fields display in the menu.
2. Optional: Deselect any criteria fields you do not want to include.

18.6.4.4.3.7 Creating break keys

Use break keys to create manageable groups of data to compare. The match set compares the data in the records within each break group only, not across the groups. Making the correct selections can save valuable processing time by preventing widely divergent data from being compared.

Break keys are especially important when you deal with large amounts of data, because the size of the break groups can affect processing time. Even if your data is not extensive, break groups will help to speed up processing.

Create break keys that group similar data that would most likely contain matches. Keep in mind that records in one break group will not be compared against records in any other break group.

For example, when you match to find duplicate addresses, base the break key on the postcode, city, or state to create groups with the most likely matches. When you match to find duplicate individuals, base the break key on the postcode and a portion of the name as the most likely point of match.

To create break keys:

1. In the How many fields column, select the number of fields to include in the break key.
2. For each break key, select the following:
 - the field(s) in the break key
 - the starting point for each field
 - the number of positions to read from each field
3. After you define the break keys, do one of the following:
 - Click [Finish](#). This completes the match transform.
 - If you are performing multi-national matching, click [Next](#) to go to the Matching Criteria page.

18.6.4.4.4 After setup

Although the Match wizard does a lot of the work, there are some things that you must do to have a runnable match job. There are also some things you want to do to refine your matching process.

Connect to downstream transforms

When the Match wizard is complete, it places the generated transforms on the workspace, connected to the upstream transform you selected to start the Match wizard. For your job to run, you must connect each port from the last transform to a downstream transform. To do this, click a port and drag to connect to the desired object.

View and edit the new match transform

To see what is incorporated in the transform(s) the Match Wizard produces, right-click the transform and choose [Match Editor](#).

View and edit Associate transforms

To see what is incorporated in the Associate transform(s) the Match Wizard produces, right-click the transform and choose [Associate Editor](#).

Multinational matching

For the Multinational consumer match strategy, the wizard builds a Match transform for each track that you create.

⚠ Caution

If you delete any tracks from the workspace after the wizard builds them, you must open the Case transform and delete any unwanted rules.

Related Information

[Unicode matching \[page 476\]](#)

18.6.4.5 Transforms for match data flows

The Match and Associate transforms are the primary transforms involved in setting up matching in a data flow. These transforms perform the basic matching functions.

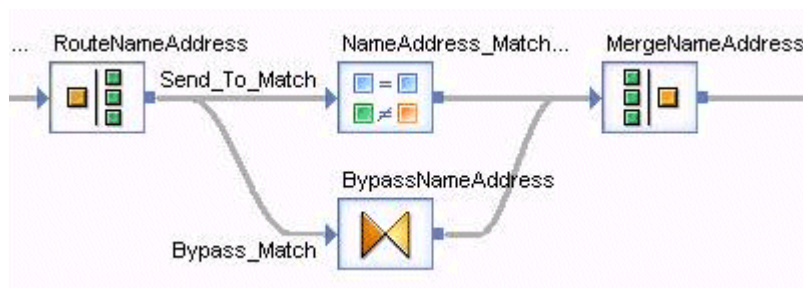
There are also other transforms that can be used for specific purposes to optimize matching.

Transform	Usage
Case	Routes data to a particular Match transform (match set). A common usage for this transform is to send USA-specific and international-specific data to different transforms. You can also use this transform to route blank records around a Match transform.

Transform	Usage
Merge	Performs the following functions: - Brings together data from Match transforms for Association matching. - Brings together matching records and blank records after being split by a Case transform.
Query	Creates fields, performs functions to help prepare data for matching, orders data, and so on.

Example

Any time you need to bypass records from a particular match process (usually in Associative data flows and any time you want to have records with blank data to bypass a match process) you will use the Case, Query, and Merge transforms.



- The Case transform has two routes: one route sends all records that meet the criteria to the Match transform, and one that sends all other records to the bypass match route.
- The Query transform adds the fields that the Match transform generates and that you output. (The output schema in the Match transform and the output schema in the Query transform must be identical for the routes to be merged.) The contents of the newly added fields in the Query transform may be populated with an empty string.
- The Merge transform merges the two routes into a single route.

18.6.4.5.1 Removing matching from the Match transform

You may want to place a transform that employs some of the functionality of a Match transform in your data flow, but does not include the actual matching features. For example, you may want to do candidate selection or prioritization in a data flow or a location in a data flow that doesn't do matching at all.

- Right-click the Match transform in the object library, and choose New.
- In the *Format name* field, enter a meaningful name for your transform. It's helpful to indicate which type of function this transform will be performing.
- Click *OK*.
- Drag and drop your new Match transform configuration onto the workspace and connect it to your data flow.

5. Right-click the new transform, and choose [Match Editor](#).
6. Deselect the [Perform matching](#) option in the upper left corner of the Match editor.

Now you can add any available operation to this transform.

18.6.4.6 Working in the Match and Associate editors

Editors

The Match and Associate transform editors allow you to set up your input and output schemas. You can access these editors by double-clicking the appropriate transform icon on your workspace.

The Match and Associate editors allow you to configure your transform's options. You can access these editors by right-clicking the appropriate transform and choosing [Match Editor](#) (or [Associate Editor](#)).

Order of setup

→ Tip

The order that you set up your Match transform is important!

First, it is best to map your input fields. If you don't, and you add an operation in the Match editor, you may not see a particular field you want to use for that operation.

Secondly, you should configure your options in the Match editor before you map your output fields. Adding operations to the Match transform (such as Unique ID and Group Statistics) can provide you with useful Match transform-generated fields that you may want to use later in the data flow or add to your database.

→ Remember

Make sure that you:

1. Map your input fields.
2. Configure the options for the transform.
3. Map your output fields.

18.6.4.7 Physical and logical sources

Track physical and logical sources for each record by designating a field that contains source identification for each record.

Tracking your data sources is essential for producing informative match reports. Data sources can be based on an input source or based on some data element in the rows that the transform reads. Track the source types described in the following table.

Source Type	Description
Physical	File name or value attributed to the source of the input data. Create the field using a Query transform or a User-Defined transform, if your input source doesn't already have a source ID field. Each record in the source contains the physical source field. When you look at statistics that result from processing many sources, you can identify the source by looking at the physical source field for each record in your statistics and match reports.
Logical	A group of records that span multiple input sources or a subset of records from a single input source. Create a source ID field using a Query transform or a User-Defined transform, when your input source doesn't already have a source ID field. This field tracks the various sources within a Reader for reporting purposes. The transform uses the field in the Group Statistics operation of the Match transform to generate the source statistics. The transform can also use the source ID field in compare tables, so that you can specify sources to compare.

18.6.4.7.1 Logical input source

Track various sources of input data for reports, statistics, and compare tables.

If you want to count source statistics in the Match transform (for the Match Source Statistics Summary report, for example), you must create a field using a Query transform or a User-Defined transform, if you don't already have one in your input data sources.

This field tracks the various sources within a Reader for reporting purposes, and is used in the Group Statistics operation of the Match transform to generate the source statistics. It is also used in compare tables, so that you can specify which sources to compare.

18.6.4.7.2 Physical input source

You track your input data source by assigning the physical source a value in a field.

Then you use this field in the transforms where report statistics are generated.

To assign this value, add a Query transform after the source and add a column with a constant containing the name you want to assign to this source.

i Note

If your source is a flat file, you can use the *Include file name* option to automatically generate a column containing the file name.

18.6.4.7.3 Using sources

A source is the grouping of records on the basis of some data characteristic that you can identify. A source might be all records from one input file, or all records that contain a particular value in a particular field.

Sources are abstract and arbitrary—there is no physical boundary line between sources. Source membership can cut across input files or database records as well as distinguish among records within a file or database, based on how you define the source.

If you are willing to treat all your input records as normal, eligible records with equal priority, then you do not need to include sources in your job.

Typically, a match user expects some characteristic or combination of characteristics to be significant, either for selecting the best matching record, or for deciding which records to include or exclude from a mailing list, for example. Sources enable you to attach those characteristics to a record, by virtue of that record's membership in its particular source.

Before getting to the details about how to set up and use sources, here are some of the many reasons you might want to include sources in your job:

- To give one set of records priority over others. For example, you might want to give the records of your house database or a suppression source priority over the records from an update file.
- To identify a set of records that match suppression sources, such as the DMA.
- To set up a set of records that should not be counted toward multi-source status. For example, some mailers use a seed source of potential buyers who report back to the mailer when they receive a mail piece so that the mailer can measure delivery. These are special-type records.
- To save processing time, by canceling the comparison within a set of records that you know contains no matching records. In this case, you must know that there are no matching records within the source, but there may be matches among sources. To save processing time, you could set up sources and cancel comparing within each source.
- To get separate report statistics for a set of records within an source, or to get report statistics for groups of sources.
- To protect a source from having its data overwritten by a best record or unique ID operation. You can choose to protect data based on membership in a source.

18.6.4.7.4 Source types

Identify a source as one of three types: Normal, suppression, or special.

The Match transform process your records differently depending on their source type.

Source	Description
Normal	A group of records that you consider to be good, eligible records.
Suppress	A grouping of records based on a specific data characteristic that you can identify. For example: • DMA Mail Preference File • American Correctional Association prisons/jails sources • No pandering or non-responder sources • Credit card or bad-check suppression sources
Special	Like a Normal source, with one exception. The transform does not count a Special source when determining whether a match group is single-source or multi-source. A Special source contributes records, but the transform does not count the records toward multi-source status.

❖ Example

ABC, Inc uses a source that contains seed names. Seed names are for people who report to the company when they receive advertisements from the company in the mail. ABC, Inc uses the information to measure mail delivery timing.

The reason for identifying the source type is so the transform can set that identity for each of the records that are members of the source. Source type plays an important role in how the transform processes matching records, like members of a match group. And also how the transform produces output, such as whether it includes or excludes a record from the output.

18.6.4.7.4.1 Manually defining input sources

Once you have mapped in an input field that contains the source values, you can create your sources in the Match Editor.

1. In the Match Editor, select *Transform Options* in the explorer pane on the left, click the *Add* button, and select *Input Sources*.
The new Input Sources operation appears under Transform Options in the explorer pane. Select it to view Input Source options.
2. In the *Value field* drop-down list, choose the field that contains the input source value.
3. In the *Define sources* table, create a source name, type a source value that exists in the Value field for that source, and choose a source type.
4. Choose value from the *Default source name* option. This name will be used for any record whose source field value is blank.

Click the *Apply* button to save any changes you have made, before you move to another operation in the Match Editor.

18.6.4.7.4.2 Automatically defining input sources

To avoid manually defining your input sources, you can choose to do it automatically by choosing the [Auto generate sources](#) option in the Input Sources operation.

1. In the Match Editor, select [Transform Options](#) in the explorer pane on the left, click the [Add](#) button, and select [Input Sources](#).
The new Input Sources operation appears under Transform Options in the explorer pane. Select it to view Input Source options.
2. In the [Value field](#) drop-down list, choose the field that contains the input source value.
3. Choose value from the [Default source name](#) option. This name will be used for any record whose source field value is blank.
4. Select the [Auto generate sources](#) option.
5. Choose a value in the [Default type](#) option
The default type will be assigned to to any source that does not already have the type defined in the Type field.
6. Select a field from the drop-down list in the [Type field](#) option.

Auto generating sources will create a source for each unique value in the Value field. Any records that do not have a value field defined will be assigned to the default source name.

18.6.4.7.5 Source groups

The source group capability adds a higher level of source management.

Source groups primarily affect reports. However, you can also use source groups to select multi-source records based on the number of source groups in which a name occurs.

❖ Example

You rent several files from two brokers. Then you define five sources for the transform to use in ranking the records. You want the transform to generate statistics broken down by broker and by file. To break down statistics by broker, you define groups of sources for each broker.

You cannot use source groups in the same way you use sources. For example, you cannot give one source group priority over another.

18.6.4.7.5.1 Creating source groups

You must have input sources in an Input Source operation defined to be able to add this operation or define your source groups.

1. Select a Match transform in your data flow, and choose [Tools > Match Editor](#).
2. In the Match Editor, select [Transform Options](#) in the explorer pane on the left, click the [Add](#) button, and select [Source Groups](#).

The new Source Groups operation appears under Input Sources operation in the explorer pane. Select it to view Source Group options.

3. Confirm that the input sources you need are in the Sources column on the right.
4. Double-click the first row in the Source Groups column on the left, and enter a name for your first source group, and press Enter.
5. Select a source in the Sources column and click the [Add](#) button.
6. Choose a value for the [Undefined action](#) option.
This option specifies the action to take if an input source does not appear in a source group.
7. If you chose Default as the undefined action in the previous step, you must choose a value in the [Default source group](#) option.
This option is populated with source groups you have already defined. If an input source is not assigned to a source group, it will be assigned to this default source group.
8. If you want, select a field in the [Source group field](#) option drop-down list that contains the value for your source groups.

18.6.4.8 Match preparation

18.6.4.8.1 Prepare data for matching

To achieve the best possible match results, process your data with other transform types to prepare data before match processing.

The following table describes several pre match data preparation methods.

Method	Description
Data correction and standardization	Accurate matches depend on good data coming into the Match transform. For batch matching, we recommend that you perform address and data cleansing before you input data to the Match transform.
Filter empty records	To help match performance, filter empty records out of your databases before you perform matching.
Noise words	Remove or replace words that are meaningless to the matching process. For example, for a firm match job, remove words such as Inc., Corp., or Ltd.
Remove punctuation	To maximize your matching process, map the following Data Cleanse transform output fields into your Match transform. The Data Cleanse transform outputs standardized data, such as all upper case with punctuation removed.

Method	Description
Break groups	To improve match performance, use break groups, which are smaller groups of records. To create a break group, you sort and combine into smaller groups records that are more likely to match. Comparing one million records takes a lot more time than comparing 100 records. Break groups organize records into collections that are potential matches, which reduce the number of comparisons that the Match transform performs. For example, create a break group that contains only records from a particular ZIP code or the first three digits of a ZIP code. i Note Create break groups before you input data into the Match transform.
Match standards	Establish match standards before you perform matching. For example, include variations of name or firm data in your match standards to help ensure a match. Set a standard that a variation of Bill is William. When creating match standards, use the original data and one or more variations. You can add anywhere from one to five variations or match standards, depending on the type of data. For example, if the transform compares first names but determines they do not match, it then compares the variations. If the variations match, the transform continues trying to match the two records instead of failing them, because it did not consider the original first names a match.
Custom Match Standards	Create custom match standards by using Data Cleanse. Match custom Data Cleanse output fields and associated aliases. Then map the custom output fields from Data Cleanse into the Match transform.

18.6.4.8.1.1 Fields to include for matching

To take advantage of the wide range of features in the Match transform, map a number of input fields in addition to the fields you map for match criteria.

The following table contains content for other fields to include in your input field mapping.

Field contents	Contains
Logical source	A value that specifies from which logical source a record originated. This field is used in the Group Statistics operation and in compare tables.

Field contents	Contains
Physical source	A value that specifies which physical source a record originated. For example, a source object, or a group of candidate-selected records. The software uses physical source fields in the Match transform options, Candidate Selection operation, and the Associate transform.
Break keys	The fields that contain the break key value for creating break groups. Including a field that already contains the break key value could help improve the performance of break group creation. A field with the break key value saves the Match transform from doing the parsing of multiple fields to create the break key.
Criteria fields	The fields that contain the data you want to match on.
Count flags	Contain a Yes or No value to specify whether a logical source should be counted in a Group Statistics operation.
Record priority	A value that signifies a record as having priority over another when ordering records. Use this field in Group Prioritization operations.
Apply blank penalty	A Yes or No value to specify whether Match should apply a blank penalty to a record. Use this field in Group Prioritization operations.
Starting unique ID value	A starting ID value that increments by 1 every time the transform assigns a unique ID. Use this field in the Unique ID operation.

The table is not a complete list. Depending on the features you want to use, you may include many other fields for the Match transform.

18.6.4.8.2 Control record comparisons

When you control record comparisons, you save processing time and increase the match quality.

When you control the actual number of comparisons, you can reduce processing times. Additionally, when you group only records that have a potential to match, Match returns better results.

Use the Group Forming section of the Match editor and configure the following operations:

- **Break groups:** Break up your records into smaller groups of records that are more likely to match.
- **Candidate selection:** Select only match candidates from a database table. Applicable for real-time jobs.

You can also use compare tables to include or exclude records for comparison by logical source.

Related Information

[Break groups and break keys \[page 420\]](#)

[Candidate selection \[page 422\]](#)

18.6.4.8.2.1 Break groups and break keys

Break groups contain records that are likely to match.

A common scenario for break groups is to create groups based on the same postcode. The chances of the Match transform finding a matching record with a different postcode are small. Therefore, a postcode break group ensures that the Match transform will never waste time comparing records from different postcodes.

Break keys

You form break groups by creating a break key. A break key is a field that consists of parts of other fields, or a single field, which is then used to group together records based on similar data.

❖ Example

The following table illustrates a typical break key that consists of the following elements combined:

- Five digits of the *Postcode1* field
- First three characters of the *Address_Primary_Name* field.

Field (Start pos:length)	Data in field	Generated break key
<i>Postcode1</i> (1:5)	10101	10101Mai
<i>Address_Primary_Name</i> (1:3)	Main	

The Match transform places all records that match the generated break key in the same break group and compares each record against one another.

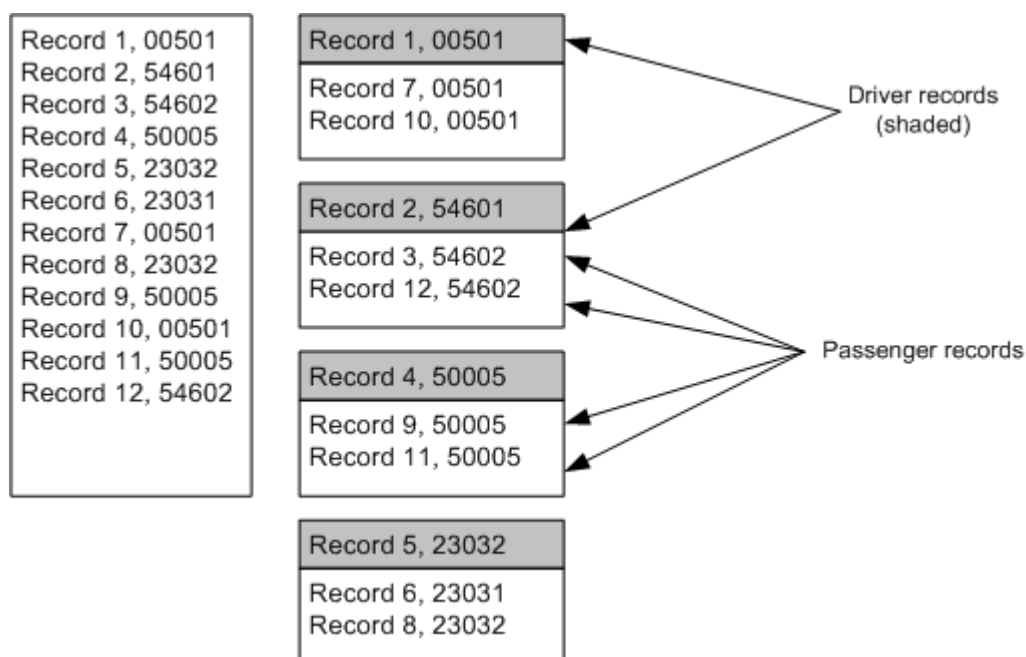
Break group anatomy

Break groups consist of driver and passenger records. The driver record is the first record in the break group, and all other records are passengers.

The driver record is the record that drives the comparison process in matching. The driver is compared to all of the passengers first.

❖ Example

The break key in the following illustration consists of the first three digits of the Postcode.



Record sorting in the break group

The transform sorts records based on the break key field.

You can add a Group Prioritization operation after the Break Groups operation to specify the driver records.

→ Remember

Order is important! If you are creating break groups using records from a Suppress-type source, be sure that the suppression records are the drivers in the break group.

Phonetic break keys

Use the Soundex and Double Metaphone functions to create fields containing phonetic codes, which can then be used to form break groups for matching.

Related Information

[Phonetic match criteria \[page 478\]](#)

[Management Console Guide: Data Quality Reports, Match Contribution report](#)

18.6.4.8.2.1.1 Creating break groups

We recommend that you standardize your data before you create your break keys. Data can be treated differently that is inconsistently cased, for example.

1. Add a Break Groups operation to the [Group Forming](#) option group.
2. in the [Break key table](#), add a row by clicking the Add button.
3. Select a field in the [field](#) column that you want to use as a break key.
Postcode is a common break key to use.
4. Choose the start position and length (number of characters) you want used in your break key.
You can use negative integers to signify that you want to start at the end of the actual string length, not the specified length of the field. For example, Field(-3,3) takes the last 3 characters of the string, whether the string has length of 10 or a length of 5.
5. Add more rows and fields as necessary.
6. Order your rows by selecting a row and clicking the [Move Up](#) and [Move Down](#) buttons.
Ordering your rows ensures that the fields are used in the right order in the break key.

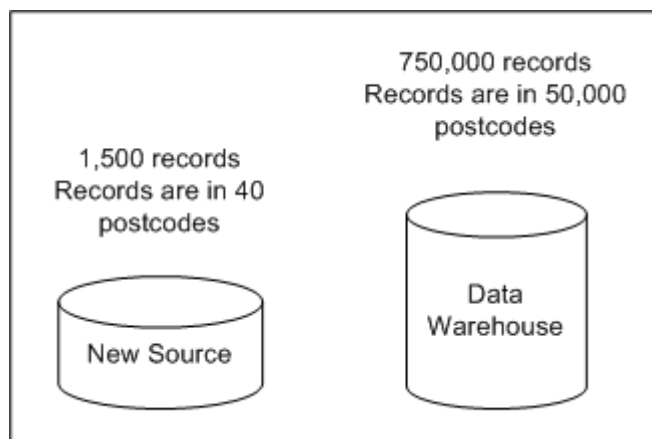
Your break key is now created.

18.6.4.8.2.2 Candidate selection

To speed processing in a match job, use the Candidate Selection operation (Group forming option group) in the Match transform to append records from a relational database to an existing data collection before matching. When the records are appended, they are not logically grouped in any way. They are simply appended to the end of the data collection on a record-by-record basis until the collection reaches the specified size.

For example, suppose you have a new source of records that you want to compare against your data warehouse in a batch job. From this warehouse, you can select records that match the break keys of the new source. This helps narrow down the number of comparisons the Match transform has to make.

For example, here is a simplified illustration: Suppose your job is comparing a new source database—a smaller, regional file—with a large, national database that includes 15 records in each of 43,000 or so postcodes. Further assume that you want to form break groups based only on the postcode.



Notes	Regional	National	Total
Without candidate selection, the Match transform reads all of the records of both databases.	1,500	750,000	751,500
With candidate selection, only those records that would be included in a break group are read.	1,500	About 600 (40 x 15)	2,100

18.6.4.8.2.2.1 Datastores and candidate selection

To use candidate selection, you must connect to a valid datastore. You can connect to any SQL-based or persistent cache datastore. There are advantages for using one over the other, depending on whether your secondary source is static (it isn't updated often) or dynamic (the source is updated often).

Persistent cache datastores

Persistent cache is like any other datastore from which you can load your candidate set. If the secondary source from which you do candidate selection is fairly static (that is, it will not change often), then you might want consider building a persistent cache, rather than using your secondary source directly, to use as your secondary table. You may improve performance.

You may also encounter performance gains by using a flat file (a more easily searchable format than a RDBMS) for your persistent cache. If the secondary source is not an RDBMS, such as a flat file, you cannot use it as a "datastore". In this case, you can create a persistent cache out of that flat file source and then use that for candidate selection.

i Note

A persistent cache used in candidate selection must be created by a data flow in double-byte mode. To do this, you will need to change the locale setting in the Data Services Locale Selector (set the code page to utf-8). Run the job to generate the persistent cache, and then you can change the code page back to its original setting if you want.

Cache size

Performance gains using persistent cache also depend on the size of the secondary source data. As the size of the data loaded in the persistent cache increases, the performance gains may decrease. Also note that if the original secondary source table is properly indexed and optimized for speed then there may be no benefit in creating a persistent cache (or even pre-load cache) out of it.

Related Information

[Persistent cache datastores \[page 181\]](#)

18.6.4.8.2.2.2 Auto-generation vs. custom SQL

There are cases where the Match transform can generate SQL for you, and there are times where you must create your own SQL. This is determined by the options you and how your secondary table (the table you are selecting match candidates from) is set up.

Use this table to help you determine whether you can use auto-generated SQL or if you must create your own.

i Note

In the following scenarios, “input data” refers to break key fields coming from a transform upstream from the Match transform (such as a Query transform) or a break key fields coming from the Break Group operation within the Match transform itself.

Scenario	Auto-generate or Custom?
You have a single break key field in your input data, and you have the same field in your secondary table.	Auto-generate
You have multiple break key fields in your input data, and you have the same fields in your secondary table.	Auto-generate
You have multiple break key fields in your input data, and you have one break key field in your secondary table.	Auto-generate
You have a single break key field in your input data, and you have multiple break key fields in your secondary table.	Custom
You have multiple break key fields in your input data, but you have a different format or number of fields in your secondary table.	Custom
You want to select from multiple input sources.	Custom

18.6.4.8.2.2.3 Break keys and candidate selection

We recommend that you create a break key column in your secondary table (the table that contains the records you want to compare with the input data in your data flow) that matches the break key you create your break groups with in the Match transform. This makes setup of the Candidate Selection operation much easier. Also, each of these columns should be indexed.

We also recommend that you create and populate the database you are selecting from with a single break key field, rather than pulling substrings from database fields to create your break key. This can help improve the performance of candidate selection.

i Note

Records extracted by candidate selection are appended to the end of an existing break group (if you are using break groups). So, if you do not reorder the records using a Group Prioritization operation after the

Candidate Selection operation, records from the original source will always be the driver records in the break groups. If you are using candidate selection on a Suppress source, you will need to reorder the records so that the records from the Suppress source are the drivers.

18.6.4.8.2.2.4 Setting up candidate selection

If you are using Candidate selection for a real-time job, be sure to deselect the [Split records into break groups](#) option in the Break Group operation of the Match transform.

To speed processing in a real-time match job, use the Candidate Selection operation (Group forming option group) in the Match transform to append records from a relational database to an existing data collection before matching. When the records are appended, they are not logically grouped in any way. They are simply appended to the end of the data collection on a record-by-record basis until the collection reaches the specified size.

1. In the Candidate Selection operation, select a valid datastore from the [Datastore](#) drop-down list.
2. In the [Cache type](#) drop-down list, choose from the following values:

Option	Description
No_Cache	Captures data at a point in time. The data doesn't change until the job restarts.
Pre-load Cache	Use this option for static data.

3. Depending on how your input data and secondary table are structured, do one of the following:
 - Select [Auto-generate SQL](#). Then select the [Use break column from database](#) option, if you have one, and choose a column from the [Break key field](#) drop-down list.

i Note

If you choose the [Auto-generate SQL](#) option, we recommend that you have a break key column in your secondary table and select the [Use break column from database](#) option. If you don't, the SQL that is created could be incorrect.

- Select [Create custom SQL](#), and either click the [Launch SQL Editor](#) button or type your [SQL](#) in the SQL edit box.
4. If you want to track your records from the input source, select [Use constant source value](#).
 5. Enter a value that represents your source in the [Physical source value](#) option, and then choose a field that holds this value in the [Physical source field](#) drop-down list.
 6. In the [Column mapping](#) table, add as many rows as you want. Each row is a field that will be added to the collection.
 - a. Choose a field in the [Mapped name](#) column.
 - b. Choose a column from your secondary table (or from a custom query) in the [Column name](#) option that contains the same type of data as specified in the [Mapped name](#) column.

If you have already defined your break keys in the Break Group option group, the fields used to create the break key are posted here, with the Break Group column set to YES.

18.6.4.8.2.2.5 Writing custom SQL

Use placeholders

To avoid complicated SQL statements, you should use placeholders (which are replaced with real input data) in your WHERE clause.

For example, let's say the customer database contains a field called MatchKey, and the record that goes through the cleansing process gets a field generated called MATCH_KEY. This field has a placeholder of [MATCHKEY]. The records that are selected from the customer database and appended to the existing data collection are those that contain the same value in MatchKey as in the transaction's MATCH_KEY. For this example, let's say the actual value is a 10-digit phone number.

The following is an example of what your SQL would look like with an actual phone number instead of the [MATCHKEY] placeholder.

```
SELECT ContactGivenName1, ContactGivenName2, ContactFamilyName, Address1,
Address2, City, Region, Postcode, Country, AddrStreet, AddrStreetNumber,
AddrUnitNumber
FROM TblCustomer
WHERE MatchKey = '123-555-9876';
```

⚠ Caution

You must make sure that the SQL statement is optimized for best performance and will generate valid results. The Candidate Selection operation does not do this for you.

Replace placeholder with actual values

After testing the SQL with actual values, you must replace the actual values with placeholders ([MATCHKEY], for example).

Your SQL should now look similar to the following.

```
SELECT ContactGivenName1, ContactGivenName2, ContactFamilyName, Address1,
Address2, City, Region, Postcode, Country, AddrStreet, AddrStreetNumber,
AddrUnitNumber
FROM TblCustomer
WHERE MatchKey = [MATCHKEY];
```

i Note

Placeholders cannot be used for list values, for example in an IN clause:

```
WHERE status IN ([status])
```

If [status] is a list of values, this SQL statement will fail.

18.6.4.8.2.3 Compare tables

Use compare tables to compare records within sources, across sources, or using a combination of both.

Compare tables are sets of rules that define what records to compare. Compare tables are almost an additional way to create break groups. You use your logical source values to determine what records the Match transform compares or does not compare.

The transform uses a compare table to determine what record pairs qualify for comparisons, and what sources to compare. If you do not include a Compare table in the Match transform, the transform uses a driver record to compare with all remaining passenger records in the data collection.

If you have many physical or logical sources in your project, it may be easier to specify what *not* to compare, as opposed to what to compare.

❖ Example

For example, you have 10 sources named A through J. You want to compare all but A and B. Set the `DEFAULT_ACTION` option to `COMPARE`. Then set up a table row for both source A and source B. Set the Action options for source A and B to `NO_MATCH`.

18.6.4.8.2.3.1 Setting up a compare table

Be sure to include a field that contains a logical source value before you add a Compare table operation to the Match transform (in the Match level option group).

Here is an example of how to set up your compare table. Suppose you have two IDs (A and B), and you only want to compare across sources, not within the sources.

1. If no Compare Table is present in the Matching section, right-click ► **Matching** ► **<Level Name>** ►, and select ► **Add** ► **Compare** ►.
2. Set the **Default action** option to **No_Match**, and type **None** in the **Default logical source value** option. This tells the Match transform to not compare everything, but follow the comparison rules set by the table entries.

i Note

Use care when choosing logical source names. Typing "None" in the **Default logical source value** option will not work if you have a source ID called "None."

3. In the **Compare actions** table, add a row, and then set the **Driver value** to **A**, and set the **Passenger value** to **B**.
4. Set **Action** to **Compare**.

i Note

Account for all logical source values. The example values entered above assumes that A will always be the driver ID. If you expect that a driver record has a value other than A, set up a table entry to account for that value and the passenger ID value. Remember that the driver record is the first record read in a collection.

If you leave the *Driver value* or *Passenger value* options blank in the compare table, then it will mean that you want to compare all sources. So a Driver value of A and a blank passenger record with an action of compare will make a record from A compare against all other passenger records.

Sometimes data in collections can be ordered (or not ordered, as the case may be) differently than your compare table is expecting. This can cause the matching process to miss duplicate records.

In the example, the way you set up your Compare action table row means that you are expecting that the driver record should have a driver value of A, but if the driver record comes in with a value of B, and the passenger comes in with a value of A, it won't be compared.

To account for situations where a driver record might have a value of B and the passenger a value of A, for example, include another row in the table that does the opposite. This will make sure that any record with a value of A or B is compared, no matter which is the Driver or Passenger.

i Note

In general, if you use a suppress source, you should compare within the other sources. This ensures that all of the matches of those sources are suppressed when any are found to duplicate a record on the suppress source, regardless of which record is the driver record.

18.6.4.8.3 Ordering and prioritizing records

You may have data sources, such as your own data warehouse, that you might trust more than records from another source, such as a rented source, for example. You may also prefer newer records over older records, or more complete records over those with blank fields. Whatever your preference, the way to express this preference in the matching process is using priorities.

There are other times where you might want to ensure that your records move to a given operation, such as matching or best record, for example, in a particular order. For example, you might want your match groups to be ordered so that the first record in is the newest record of the group. In this case, you would want to order your records based on a date field.

Whatever the reason, there are two ways to order your records, either before or after the comparison process:

- Sorting records in break groups or match groups using a value in a field.
- Using penalty scores. These can be defined per field, per record, or based on input source membership.

Match editor

You can define your priorities and order your records in the Group Prioritization operation, available in Group Forming and in the Post-match processing operations of each match level in the Match editor.

Types of priorities

There are a couple of different types of priorities to consider:

Priority	Brief description
Record priority	Prefers records from one input source over another.
Blank penalty	Assigns a lower priority to records in which a particular field is blank.

Pre-match ordering

When you create break groups, you can set up your [Group Forming](#) [Group Prioritization](#) operation to order (or sort) on a field, besides ordering on the break key. This will ensure that the highest priority record is the first record (driver) in the break group.

You will also want to have Suppress-type input sources to be the driver records in a break group.

Post-match ordering

After the Match transform has created all of the match groups, and if order is important, you can use a Group Prioritization operation before a Group Statistics, Best Record, and Unique ID operations to ensure that the master record is the first in the match group.

→ Tip

If you are not using a blank penalty, order may not be as important to you, and you may not want to include a Group Prioritization operation before your post-match operations. However, you may get better performance out of a Best Record operation by prioritizing records and then setting the *Post only once per destination* option to Yes.

Blank penalty

Given two records, you may prefer to keep the record that contains the most complete data. You can use blank penalty to penalize records that contain blank fields.

Incorporating a blank penalty is appropriate if you feel that a blank field shouldn't disqualify one record from matching another, and you want to keep the more complete record. For example, suppose you are willing to accept a record as a match even if the Purname, Given_Name1, Given_Name2, Primary_Postfix and/or Secondary Number is blank. Even though you accept these records into your match groups, you can assign them a lower priority for each blank field.

18.6.4.8.3.1 Ordering records by sorting on a field

Be sure you have mapped the input fields into the Match transform that you want to order on, or they won't show up in the field drop-down list.

Use this method of ordering your records if you do not consider completeness of data important.

1. Enter a Prioritization name, and select the Priority Order tab.
2. In the *Priority fields* table, choose a field from the drop-down list in the *Input Fields* column.
3. In the *Field Order* column, choose *Ascending* or *Descending* to specify the type of ordering.
For example, if you are comparing a Normal source to a Suppress source and you are using a source ID field to order your records, you will want to ensure that records from the Suppress source are first in the break group.
4. Repeat step 2 for each row you added.
5. Order your rows in the *Priority fields* table by using the *Move Up* and *Move Down* buttons.

The first row will be the primary order, and the rest will be secondary orders.

18.6.4.8.3.2 Penalty scoring system

The blank penalty is a penalty-scoring system. For each blank field, you can assess a penalty of any non-negative integer.

You can assess the same penalty for each blank field, or assess a higher penalty for fields you consider more important. For example, if you were targeting a mailing to college students, who primarily live in apartments or dormitories, you might assess a higher penalty for a blank Given_Name1 or apartment number.

Field	Blank penalty
Prename	5
Given_Name1	20
Given_Name2	5
Primary Postfix	5
Secondary Number	20

As a result, the records below would be ranked in the order shown (assume they are from the same source, so record priority is not a factor). Even though the first record has blank prename, Given_Name2, and street postfix fields, we want it as the master record because it does contain the data we consider more important: Given_Name1 and Secondary Number.

Prename (5)	Given Name1 (20)	Given Name2 (5)	Family Name	Prim Range	Prim Name	Prim Post- fix (5)	Sec Num- ber (20)	Blank-field penalty
	Maria		Ramirez	100	Main		6	5 + 5 + 5 = 15
Ms.	Maria	A	Ramirez	100	Main	St		20
Ms.			Ramirez	100	Main	St	6	20 + 5 = 25

18.6.4.8.3.3 Blank penalty interacts with record priority

The record priority and blank penalty scores are added together and considered as one score.

For example, suppose you want records from your house database to have high priority, but you also want records with blank fields to have low priority. Is source membership more important, even if some fields are blank? Or is it more important to have as complete a record as possible, even if it is not from the house database?

Most want their house records to have priority, and would not want blank fields to override that priority. To make this happen, set a high penalty for membership in a rented source, and lower penalties for blank fields:

Source	Record priority (penalty points)	Field	Blank penalty
House Source	100	Given Name1	20
Rented Source A	200	Given_Name2	5
Rented Source B	300	Primary Postfix	5
Rented Source C	400	Secondary Number	20

With this scoring system, a record from the house source always receives priority over a record from a rented source, even if the house record has blank fields. For example, suppose the records below were in the same match group.

Even though the house record contains five blank fields, it receives only 155 penalty points (100 + 5 + 20 + 5 + 5 + 20), while the record from source A receives 200 penalty points. The house record, therefore, has the lower penalty and the higher priority.

Source	Given Name1	Given Name2	Family	Prim Range	Prim Name	Sec Num	Post- code	Rec pri- ority	Blank Penalty	Total
House			Smith	100	Bren		55343	100	55	155
Source A	Rita	A	Smith	100	Bren	12A	55343	200	0	200

Source	Given Name1	Given Name2	Family	Prim Range	Prim Name	Sec Num	Post-code	Rec priority	Blank Penalty	Total
Source B	Rita		Smith	100	Bren	12	55343	300	10	310

You can manipulate the scores to set priority exactly as you'd like. In the example above, suppose you prefer a rented record containing first-name data over a house record without first-name data. You could set the first-name blank penalty score to 500 so that a blank first-name field would weigh more heavily than any source membership.

18.6.4.8.3.4 Defining priority and penalty using field values

Be sure to map in any input fields that carry priority or blank penalty values.

This task tells Match which fields hold your record priority and blank penalty values for your records, and whether to apply these per record.

1. Add a Group Prioritization operation to the Group Forming or Post Match Processing section in the Match Editor.
2. Enter a Prioritization name (if necessary) and select the [Record Completeness](#) tab.
3. Select the [Order records based on completeness of data](#) option.
4. Select the [Define priority and penalty fields](#) option.
 - [Define only field penalties](#): This option allows you to select a default record priority and blank penalties per field to generate your priority score.
 - [Define priority and penalty based on input source](#): This allows you to define priority and blank penalty based on membership in an input source.
5. Choose a field that contains the record priority value from the [Record priority field](#) option.
6. In the [Apply blank penalty field](#) option, choose a field that contains the Y or N indicator for whether to apply a blank penalty to a record.
7. In the [Default record priority](#) option, enter a default record priority to use if a record priority field is blank or if you do not specify a record priority field.
8. Choose a [Default apply blank penalty](#) value (Yes or No). This determines whether the Match transform will apply blank penalty to a record if you didn't choose an apply blank penalty field or if the field is blank for a particular record.
9. In the [Blank penalty score](#) table, choose a field from the [Input Field](#) column to which you want to assign blank penalty values.
10. In the [Blank Penalty](#) column, type a blank penalty value to attribute to any record containing a blank in the field you indicated in [Input Field](#) column.

18.6.4.8.3.5 Defining penalty values by field

This task lets you define your default priority score for every record and blank penalties per field to generate your penalty score.

1. Add a Group Prioritization operation to the Group Forming or Post Match Processing section in the Match Editor.
2. Enter a Prioritization name (if necessary) and select the [Record Completeness](#) tab.
3. Select the [Order records based on completeness of data](#) option.
4. Select the [Define only field penalties](#) option.
5. In the [Default record priority](#) option, enter a default record priority that will be used in the penalty score for every record.
6. Choose a [Default apply blank penalty](#) value (Yes or No). This determines whether the Match transform will apply blank penalty to a record if you didn't choose an apply blank penalty field or if the field is blank for a particular record.
7. In the [Blank penalty score](#) table, choose a field from the [Input Field](#) column to which you want to assign blank penalty values.
8. In the [Blank Penalty](#) column, type a blank penalty value to attribute to any record containing a blank in the field you indicated in [Input Field](#) column.

18.6.4.8.4 Prioritizing records based on source membership

However you prefer to prioritize your sources (by sorting a break group or by using penalty scores), you will want to ensure that your suppress-type source records are the drivers in the break group and comparison process.

For example, suppose you are a charitable foundation mailing a solicitation to your current donors and to names from two rented sources. If a name appears on your house source and a rented source, you prefer to use the name from your house source.

For one of the rented sources, Source B, suppose also that you can negotiate a rebate for any records you do not use. You want to use as few records as possible from Source B so that you can get the largest possible rebate. Therefore, you want records from Source B to have the lowest preference, or priority, from among the three sources.

Source	Priority
House source	Highest
Rented source A	Medium
Rented source B	Lowest

Suppress-type sources and record completeness

In cases where you want to use penalty scores, you will want your Suppress-type sources to have a low priority score. This makes it likely that normal records that match a suppress record will be subordinate matches in a match group, and will therefore be suppressed, as well. Within each match group, any record with a lower priority than a suppression source record is considered a suppress match.

For example, suppose you are running your files against the DMA Mail Preference File (a list of people who do not want to receive advertising mailings). You would identify the DMA source as a suppression source and assign a priority of zero.

Source	Priority
DMA Suppression source	0
House source	100
Rented source A	200
Rented source B	300

Suppose Match found four matching records among the input records.

Matching record (name fields only)				Source	Priority
	Maria		Ramirez	House	100
Ms.			Ramirez	Source B	300
Ms.	Maria	A	Ramirez	Source A	200
Ms.	Maria	A	Ramirez	DMA	0

The following match group would be established. Based on their priority, Match would rank the records as shown. As a result, the record from the suppression file (the DMA source) would be the master record, and the others would be subordinate suppress matches, and thus suppressed, as well.

Source	Priority
DMA	0 (Master record)
House	100
Source A	200
Source B	300

18.6.4.8.4.1 Defining penalties based on source membership

In this task, you can attribute priority scores and blank penalties to an input source, and thus apply these scores to any record belonging to that source. Just be sure you have your input sources defined before you attempt to complete this task.

1. Add a Group Prioritization operation to the Group Forming or Post Match Processing section in the Match Editor.
2. Enter a Prioritization name (if necessary) and select the [Record Completeness](#) tab.
3. Select the [Order records based on completeness of data](#) option.
4. Select the [Define priority and penalty based on input source](#) option.
5. In the [Source Attributes](#) table, select a source from the drop-down list.
6. Type a value in the [Priority](#) column to assign a record priority to that source.

Remember that the lower the score, the higher the priority. For example, you would want to assign a very low score (such as 0) to a suppress-type source.

7. In the [Apply Blank Penalty](#) column, choose a Yes or No value to determine whether to use blank penalty on records from that source.
8. In the [Default record priority](#) option, enter a default record priority that will be used in the penalty score for every record that is not a member of a source.
9. Choose a [Default apply blank penalty](#) value (Yes or No). This determines whether to apply blank penalties to a record that is not a member of a source.
10. In the [Blank penalty score](#) table, choose a field from the [Input Field](#) column to which you want to assign blank penalty values.
11. In the [Blank Penalty](#) column, type a blank penalty value to attribute to any record containing a blank in the field you indicated in [Input Field](#) column.

18.6.4.8.5 Data Salvage

Data salvaging prevents the Match transform from making matches between records that you do not want to match.

With data salvaging, the transform temporarily copies data from a passenger record to the driver record after it compares the two records. The data that's copied is data in the passenger record that is missing or incomplete in the driver record.

❖ Example

If you do not enable data salvaging, the transform includes the records in the following table in the same match group. The driver record contains a blank Name field, but the Address and Postcode fields match.

Record	Name	Address	Postcode
1 (driver)		123 Main St.	54601
2	John Smith	123 Main St.	54601
3	Jack Hill	123 Main St.	54601

If you enable data salvaging, the transform temporarily copies the Name from record 2, John Smith, to the driver record. The transform does not make a match group from the three records: Record 1 matches Record 2, but Record 1 does not match Record 3 because John Smith doesn't match Jack Hill.

Record	Name	Address	Postcode
1 (driver)	John Smith (copied from record 2)	123 Main St.	54601
2	John Smith	123 Main St.	54601
3	Jack Hill	123 Main St.	54601

❖ Example

You have a suppression source, which is a list of no-pandering addresses. You set the suppression source with the highest priority, and you do not enable data salvaging. The transform suppresses all records that

match the suppression source records. A suppression record of 123 Main St matches 123 Main St #2 and 123 Main St Apt C. Therefore, the transform suppresses both addresses.

18.6.4.8.5.1 Data salvaging and initials

When a driver record has a Name field with an initial instead of a full name, the Match transform may temporarily borrow the full name of a matching record for data salvaging.

By retaining more information for the driver record, data salvaging helps improve the quality of your matching results.

Note

The transform performs data salvaging for initials only with the given name and family name fields.

Example

The following three records represent potential matches. The transform grouped the records into a break group based on the address and ZIP Code (not shown).

Record	First name	Last name	Address	Notes
357	J	L	123 Main	Driver
391	Juanita	Lopez	123 Main	
839	Joanne	London	123 Main	Lowest ranking record

The first match comparison is between the driver record, 357, and the next highest ranking record, 391. The transform calls the two records a match. Juanita and Lopez are temporarily copied to the first and last name fields of record 357.

The next match comparison is between record 357 and the next lower ranking record, 839. With data salvaging enabled, the data in the driver record now contains Juanita Lopez, borrowed from the first comparison. Therefore, the transform concludes that record 839 is not a match to record 357.

Initials and suppress-type records

If the driver record is a suppress-type record, turning off data salvaging may be beneficial.

Example

The following table contains records in a suppression source. You disable data salvaging.

Record	First name	Last name	Address	Notes
357	J	L	123 Main	Driver
391	Juanita	Lopez	123 Main	

Record	First name	Last name	Address	Notes
839	Joanne	London	123 Main	Lowest ranking record

The transform makes the first match comparison between the driver record, 357, and the next- highest ranking record, 391. The transform considers the two records a match because the JL in the driver record matches Juanita Lopez.

The transform makes the next comparison between the driver record, 357, and the next lower ranking record, 839. The transform considers the two records a match because driver JL in the driver record matches Joanne London.

Since both records 391 and 839 matched the suppress-type driver record, 357, the transform designates both records as suppress matches and does not include the records in the output.

18.6.4.8.5.2 Controlling data salvaging using a field

You can use a field to control whether data salvage is enabled. If the field's value is Y for a record, data salvaging is enabled. Be sure to map the field into the Match transform that you want to use beforehand.

1. Open the Match Editor for a Match transform.
2. In the Transform Options window, click the [Data Salvage](#) tab.
3. Select the Enable data salvage option, and choose a default value for those records.
The default value will be used in the cases where the field you choose is not populated for a particular record.
4. Select the [Specify data salvage by field](#) option, and choose a field from the drop-down menu.

18.6.4.8.5.3 Controlling data salvaging by source

You can use membership in an input source to control whether data salvage is enabled or disabled for a particular record. Be sure to create your input sources beforehand.

1. Open the Match Editor for a Match transform.
2. In the Transform Options window, click the [Data Salvage](#) tab.
3. Select the [Enable data salvage](#) option, and choose a default value for those records.
The default value will be used if a record's input source is not specified in the following steps.
4. Select the [Specify data salvage by source](#) option.
5. In the table, choose a Source and then a Perform Data Salvage value for each source you want to use.

18.6.4.9 Match criteria

18.6.4.9.1 Match criteria

Use match criteria in each match level to determine the threshold scores for matching and to define how the Match transform treats various types of data.

Match criteria represent the fields to compare. For example, to match on the first ten characters of a given name and the first 15 characters of the family name, you create two criteria that specify these requirements.

Criteria provide a way to let the Match transform know what kind of data is in the input field and, therefore, what types of operations to perform on that data.

Pre-defined vs. custom criteria

The following table details the two types of criteria supported in Match.

Criteria	Description
Predefined	Set criteria for fields that you typically use for matching such as first name, family name, and address components. With field criteria, the transform identifies the type of data in the field, and optimizes the data for matching without altering the actual input data.
Data Cleanse Custom	Set criteria for custom output fields such as user-defined and non-party data. Map the custom output fields from the Data Cleanse transform to the Match transform. The custom fields are then available in the Match Criteria input fields group. Set criteria for other types of fields such as part numbers or proprietary data for which there is no pre-defined criteria. The transform matches on abbreviations, substrings, and numeric data, but not on certain cross-field comparisons such as name match.

Match criteria pre-comparison options

Use the Address Cleanse and Data Cleanse transforms for most of the pre-match tasks to standardize data. However, the Match transform also performs some preprocessing per criteria to provide more accurate matches.

i Note

The transform doesn't actually alter your data when it performs the preprocessing.

Find the following options to control the pre-comparison standardization in the [Options](#) and [Multi Field Comparisons](#) tabs of the Match editor.

- Convert diacritical characters
- Convert text to numbers
- Convert to uppercase
- Remove punctuation
- Locale

Related Information

18.6.4.9.1.1 Adding and ordering a match criteria

You can add as many criteria as you want to each match level in your Match transform.

1. Select the appropriate match level or Match Criteria option group in the Option Explorer of the Match Editor, and right-click.
2. Choose [Criteria](#).
3. Enter a name for your criteria in the [Criteria name](#) box.
You can keep the default name for pre-defined criteria, but you should enter a meaningful criteria name if you chose a Custom criteria.
4. On the [Criteria Fields](#) tab, in the [Available criteria](#) list, choose the criteria that best represents the data that you want to match on. If you don't find what you are looking for, choose the Custom criteria.
5. In the [Criteria field mapping](#) table, choose an input field mapped name that contains the data you want to match on for this criteria.
6. Click the [Options](#) tab.
7. Configure the [Pre-comparison options](#) and [Comparison rules](#).
Be sure to set the Match score and No match score, because these are required.
8. If you want to enable multiple field (cross-field) comparison, click the [Multiple Fields Comparisons](#) tab, and select the [Compare multiple fields](#) option.
 - a. Choose the type of multiple field comparison to perform:
 - [All selected fields in other records](#): Compare each field to all fields selected in the table in all records.
 - [The same field in other records](#): Compare each field only to the same field in all records.
 - b. In the [Additional fields to compare](#) table, choose input fields that contain the data you want to include in the multiple field comparison for this criteria.

→ Tip

You can use custom match criteria field names for multiple field comparison by typing in the [Custom name](#) column.

i Note

If you enable multiple field comparison, any appropriate match standard fields are removed from the *Criteria field mapping* table on the *Criteria Fields* tab . If you want to include them in the match process, add them in the *Additional fields to compare* table.

9. Configure the *Pre-comparison* options for multiple field comparison.
10. To order your criteria in the Options Explorer of the Match Editor (or the Match Table), select a criteria and click the *Move Up* or *Move Down* buttons as necessary.

18.6.4.9.1.2 Setting up for match standards criteria

Be sure you have created and mapped in match standard fields from the Data Cleanse transform.

1. In the Match transform, add a Person1_Given_Name1 match criteria by selecting it from the Person category.
The editor automatically adds that criteria and its associated match standard criteria to the *Criteria field mapping* table.
2. Choose a field in the *Input field mapped name* column that best represents the data to be compared.

For Given_Name2 data (middle names), complete the above procedure, but use the following criteria:

- Person1_Given_Name2
- Person1_Given_Name2_Match_Std1
- Person1_Given_Name2_Match_Std2
- Person1_Given_Name2_Match_Std3

You can also use match standards for prename, maturity postname, honorary postname, firm, and firm location fields.

18.6.4.9.2 Matching methods

Matching methods are ways to set up and order match criteria to get desired matching results.

The following table contains descriptions of matching methods that you can use in the Match transform. Each of these methods has advantages and disadvantages, so consider them carefully.

Match method	Description
Rule-based	You control the criteria that the transform uses to determine a match by setting Match and No Match scores.
Weighted-scoring	Assign importance, or weight, to any criteria. The transform evaluates every rule before it determines a match, which may affect processing time.
Combination method	Use a combination of matching methods based on your data and your matching goals.

18.6.4.9.2.1 Similarity score

The similarity score is the percentage that your data is alike.

The software calculates the score internally when the Match transform compares records. Whether the software considers the records a match depends on the match and no match scores that you define in the Criteria option group (as well as other factors, but for now let's focus on these scores).

❖ Example

An example of how the transform determines similarity scores.

The following comparison table is an example. It does not illustrate how the matching process works.

Comparison	No match	Match	Similarity score	Matching?
Smith > Smith	72	95	100%	Yes
Smith > Smitt	72	95	80%	Depends on other criteria
Smith > Smythe	72	95	72%	No
Smith > Jones	72	95	20%	No

Considerations:

- The transform considers only the first comparison a match because the similarity score meets or exceeds the match score.
- In the second comparison, the transform cannot determine a match using a single criteria. In this situation, the transform moves to the next criteria, if possible.
- The transform considers the last comparison a no-match because the similarity score is less than the no-match score.

18.6.4.9.2.2 Rule-based method

In rule-based matching, the transform relies on the match and no-match scores to determine matches within the criteria.

The following example shows how to set up rule-based criteria in the Match transform.

❖ Example

Criteria	Record A	Record B	No match score	Match score	Similarity score
<i>Given Name1</i>	Mary	Mary	82	101	100
<i>Family Name</i>	Smith	Smitt	74	101	80

Criteria	Record A	Record B	No match score	Match score	Similarity score
<i>E-mail</i>	msmith@sap.com	mary.smith@sap.com	79	80	91

- For the first two criteria, *Given Name1* and *Family Name*, enter a value of 101 for the Match Score column. Setting up the Match Score with 101, the transform never considers the *Given Name1* and *Family Name* fields to determine a match. However, the transform may use the field to determine a no-match.
- A match score of 101 ensures that the criteria does not cause the records to be a match, because two fields cannot be more than 100 percent alike.
- Set the *Match Score* and *No Match Score* options for the *E-mail* criteria without a gap. This way, any comparison that reaches the last criteria must either be a match or a no-match.

→ Remember

Order is important! For performance reasons, you should have the criteria that is most likely to make the match or no-match decisions first in your order of criteria. Ordering can help reduce the number of criteria comparisons.

18.6.4.9.2.3 Weighted-scoring method

With weighted scoring, the transform relies on the total contribution score to determine matches instead of the match and no-match scores alone.

Contrast weighted scoring to the rule-based matching method. For rule-based matching, the transform gives all of the criteria the same amount of importance or weight. That is, if any criteria fails to meet the specified match score, the transform determines that the records do not match.

Contribution values

Set contribution values to assign weight to individual criteria. The higher the contribution value, the more weight that criteria carries in determining matches. Generally, the following fields carry more weight than others: Account numbers, Social Security numbers, customer numbers, Postcode1, and addresses.

i Note

Ensure that the total of all contribution values totals 100. Keep in mind, however, that you do not have to set a contribution value to all criteria.

Define contribution values in the *Contribution to Weighted Score* option in the *Criteria*.

Contribution and total contribution score

The Match transform generates the contribution score for each criteria by multiplying the contribution value you assign with the similarity score (the percentage alike). The software then adds these individual contribution scores to get the total contribution score.

Weighted match score

In the weighted scoring method, the transform determines matches only by comparing the total contribution score with the weighted match score.

- If the total contribution score is equal to or greater than the weighted match score, the transform considers the records a match.
- If the total weighted score is less than the weighted match score, the transform considers the records a no-match.

Set the weighted match score in the *Weighted match score* option of the *Level option* group. The following table is an example of how to set up weighted scoring. Notice the various types of scores that we have discussed. Also notice the following:

❖ Example

The following table is an example of how to set up weighted scoring. Notice the various types of scores that we have discussed. We considered the following information when we set the weighted scoring:

- To ensure that the transform does not find a match or a no-match, we set the *No Match Score* option to -1, and the *Match Score* option to 101.
- We set the highest contribution value to the E-mail criteria to give it the most importance.

Criteria	Record A	Record B	No match	Match	Similarity score	Contribution value	Contribution score (similarity X contribution value)
<i>First Name</i>	Mary	Mary	-1	101	100	25	25
<i>Last Name</i>	Smith	Smitt	-1	101	80	25	20
<i>E-mail</i>	ms@ sap.com	msmith@ sap.com	-1	101	84	50	42
							Total contribution score: 87

If the weighted match score is 87, then the transform considers any comparison whose total contribution score is 87 or greater a match. In this example, the comparison is a match because the total contribution score is 87.

18.6.4.9.2.4 Combination method

For the combination method, the transform combines the rule-based and weighted scoring methods of matching.

❖ Example

In the following table, we use the rule-based method and the weighted score method to determine matching records. The following table shows that the two records do not match based on the combined method.

Criteria	Record A	Record B	No match	Match	Sim score	Contribution value	Contribution score (actual similarity X contribution value)
First Name	Mary	Mary	59	101	100	25	25
Last Name	Smith	Hope	59	101	22	N/A (No Match)	N/A
E-mail	ms@ sap.com	msmith@ sap.com	49	101	N/A	N/A	N/A
							Total contribution score: N/A

18.6.4.9.3 Matching business rules

Business rules establish how you want the transform to handle various forms of and differences in your data.

An important part of the matching process is determining how you want to handle various forms of and differences in your data.

❖ Example

Every field in Record A matches every field in Record B except Record B has a blank value in one of the fields. Should the transform consider Record A and Record B as matches?

Figuring out the software behavior in this situation is a part of defining your business rules. Define most of your business rules in the [Match Criteria](#) group.

18.6.4.9.3.1 Matching on strings, abbreviations, and initials

Part of defining your business rules is to make settings for matching strings, abbreviations, and initials.

String data

Configure the transform to match longer strings to shorter strings using the *Substring adjustment score* option in the *Match Criteria* group.

❖ Example

When you configure the *Substring adjustment score*, the transform matches the string "Mayfield Painting and Sand Blasting" with "Mayfield painting."

Approximate substring

Configure the transform to match leftover portions of a string that did not match during the substring matching. Set the *Approximate substring adjustment score* and the *Substring adjustment score* options so the transform assigns a score to all of the leftover words and spaces that do not match during the substring matching.

Abbreviations

Use the *Abbreviation adjustment score* option to enable the transform to match whole words to abbreviations.

❖ Example

Transform matches "International Health Providers" to "Intl Health Providers."

Initials and acronyms

Use the *Initials adjustment score* option to enable the transform to match initials to whole words.

❖ Example

The transform matches "International Health Providers" to "IHP."

18.6.4.9.3.2 Extended abbreviation matching

Use the extended abbreviation options to enable matching on situations that are not covered by the string, substring, and initials options.

Extended abbreviation matching offers functionality that handles situations such as the following:

- You have localities information in your data such as La Crosse and New York. However, your data lists these localities both with the spaces and without the spaces as in LaCrosse and NewYork. Under normal matching, the spaces prevent the transform from matching words with spaces to the same words without spaces at 100%. Normally, the score for La Crosse and LaCrosse is 94%. The score for New York and NewYork is 93%.
- Your data has Metropolitan Life and MetLife (an abbreviation and combination of Metropolitan Life) in your data. The [Abbreviation adjustment score](#) option does not detect the combination of the two words.

If you are concerned about either of these cases in your data, you should use the [Ext abbreviation adjustment score](#) option. When you set the [Ext abbreviation adjustment score](#) option in the [Match Criteria](#) group, the transform recognizes the example situations as matches.

How the extended abbreviation adjustment score works

The score you set in the [Ext abbreviation adjustment score](#) option tunes your similarity score to consider these types of abbreviations and combinations in your data.

The adjustment score adds a penalty for the non-matched part of the words. The higher the number, the lower the penalty. A score of 100 means no penalty and score of 0 means maximum penalty.

❖ Example

The following table shows the assigned similarity score when you set the [Ext abbreviation adjustment score](#) option to 0, 50, and 100.

String 1	String 2	Sim score when Adj score is 0	Sim score when Adj score is 50	Sim score when Adj score is 100	Notes
MetLife	Metropolitan Life	58	79	100	
MetLife	Met Life	93	96	100	
MetLife	MetropolitanLife	60	60	60	The similarity score is the same for all three settings. Each string is one word, therefore, the transform did not use the Ext abbreviation adjustment score setting.

18.6.4.9.3.3 Name matching

The Match transform provides many ways to ensure that variations on names or multiple names, are considered.

Part of creating your business rules is to define how you want names handled in the matching process.

Note

Unlike other business rules, you set name matching options in the *Match Criteria* group, because they affect all appropriate name-based match criteria.

Two names; two persons

With the *Number of names that must match* option, control how the transform performs matching on match keys with more than one name. For example, comparing "John and Mary Smith" to "Dave and Mary Smith." Choose whether the transform matches only one name for the records to be identified as a match, or whether the transform disregards any persons other than the first name it parses.

With this method you can require either one or both persons to match for the record to match.

Two names; one person

With the *Compare Given_Name1 to Given_Name2* option, compare the *Given_Name1* data (first name) from one record with the *Given_Name2* data (middle name) from the second record. With this option enabled, the transform correctly identifies matching records such as the two partially shown in the following table. Typically, these record pairs represent sons or daughters named for their parents, but known by their middle name.

Example

Record #	First name	Middle name	Last name	Address
170	Leo	Thomas	Smith	225 Pushbutton Dr
198	Tom		Smith	225 Pushbutton Dr

Hyphenated family names

With the *Match on hyphenated family name* option, control how the transform performs matching when a *Family_Name* (last name) field contains a hyphenated family name. For example, comparing "Smith-Jones" to "Jones." Choose whether the transform requires both names to match, or just one of the names to match to consider the records a match.

18.6.4.9.3.1 Match compound family names

Use the *Approximate substring adjustment score* option to assist in setting up comparisons of compound family names.

The transform assigns the *Approximate substring adjustment score* to the words that do not match to other words in a compared string. The *Approximate substring adjustment score* option loosens some of the requirements of the *Substring adjustment score* option in the following ways:

- First words in a string do not have to match exactly.
- The transform uses initials and abbreviation adjustments for the words that match. (For example, Rodriguez and RDZ.)
- Matching words have to be in the same order, but there can be non-matching words before or after the matching words.
- The transform assigns the leftover words and spaces in the compared string to the *Approximate substring adjustment score*.

The *Approximate substring adjustment score* option increases the score for some matches found when using the *Substring adjustment score*.

❖ Example

The following table shows how settings in the substring options affect the calculated similarity score when the transform compares CRUZ RODRIGUEZ and GARCIA CRUZ DE RDZ.

Similarity score	Substring adjustment score	Abbreviation adjustment score	Approximate substring adjustment score
48	None	None	None
66	80	80	None
91	None	80	80

18.6.4.9.3.4 Numeric data matching

Specify how the Match transform matches data that consists of a mixture of numbers and letters.

To set numeric data requirements, use the *Numeric words match exactly* option.

Select a value so the transform requires data with a mixture of numbers and letters to match exactly in any position, or in the same position, for example. The *Numeric words match exactly* option applies most often to address data and custom data such as a part number.

The Match transform processes data with numeric and letter data as follows:

1. Breaks the string into words using punctuation and spacing.
2. Assigns a numeric attribute to the words.

i Note

A numeric word is any word that contains at least one number from 0 to 9. For example, 4L is considered a numeric word, whereas FourL is not.

3. Performs numeric matching based on settings in the *Numeric words match exactly* option.

Numeric matching option values and how they work

The following table describes the options for *Numeric words match exactly*.

Option value	Description
<i>Any_Position</i>	Requires that numeric words match exactly. The position of the word is not important. ❖ Example The transform determines that the following street addresses are a match: 4932 Main St # 101 # 101 4932 Main St The transform determines that the following street addresses are not a match: 4932 Main St # 101 # 102 4932 Main St The transform determines that the following part numbers are a match: ACCU 1.4L 29BAR ACCU 29BAR 1.4L
<i>Same_Position</i>	Requires that numeric words match exactly. This value differs from the <i>Any_Position</i> value in that the position of the word is important. ❖ Example The transform determines that, for the phone number 608-123-4567: • 608-123-4567 is a match • 123-608-4567 is not a match

Option value	Description
Any_Position_Consider_Punctuation	The transform performs word breaking on all punctuation and spaces, but not on decimal separators (period or comma). Therefore, the transform does not break any decimal numbers into words. ❖ Example The transform considers the string 123.456 as a single numeric word instead of two numeric words. The position of the numeric word is not important, however, decimal separators impact the matching process. ❖ Example The transform determines that the following strings are a match: ACCU 29BAR 1.4L ACCU 1.4L 29BAR The transform determines that the following strings are not a match because there are different decimal indicators between the 1 and the 4 in each line: ACCU 1,4L 29BAR ACCU 29BAR 1.4L The transform determines that the following strings are not a match because there are different decimal indicators between the 25 and the 435 in each line: 25,435 25.435

Option value	Description
Any_Position_Ignore_Punctuation	This value behaves similarly to the Any_Position_Consider_Punctuation value, except that decimal separators do not impact the matching process. ❖ Example The transform determines that the following strings are a match: ACCU 29BAR 1.4L ACCU 1.4L 29BAR The transform determines that the following strings are a match, even though there is a different decimal indicator between the 1 and the 4: ACCU 1,4L 29BAR ACCU 29BAR 1.4L The transform determines that the following strings are a match, even though there is a different decimal indicator between the 25 and the 435: 25,435 25.435

18.6.4.9.3.5 Blank field matching

Control how the transform compares fields when one or both of the fields are blank in the [Match Criteria](#) group.

❖ Example

Record #2 in the following table is missing a first name. For your business rules, do you want the Match transform to consider these records as matches? What if the first name field is blank in both records?

Record #1	Record #2
John	
Doe	Doe
204 Main St	204 Main St
La Crosse WI	La Crosse WI
54601	54601

The following options in the Match transform enable you to control the way the transform compares records with missing fields:

- [Both fields blank operation](#)

- *Both fields blank score*
- *One field blank operation*
- *One field blank score*

Blank field operations

The *Both fields blank operation* and *One field blank operation* options have the following value choices:

- *Eval*: The transform scores the comparison using the score you enter for the "score" options.
- *Ignore*: Ignores comparison criteria and distributes criteria contribution to the weighted score proportionally among the remaining criteria.

Blank field scores

The *Both fields blank score* and *One field blank score* options control how the Match transform scores field comparisons when the field is blank in one or both records. Enter any value from 0 to 100.

To help you decide what score to enter, determine if the Match transform should consider a blank field:

- 0% similar to a populated field or another blank field
- 100% similar to a populated field or another blank field
- Somewhere in between 0% and 100%.

Your answer probably depends on what field you're comparing. Giving a blank field a high score might be appropriate if you're matching on a first or middle name or a company name, for example.

The following examples may help you understand how your settings of the blank matching options affect the overall scoring of records.

❖ Example

One field blank operation = Ignore

Input field: *Given_Name1*

i Note

Note that when you set the blank options to *Ignore*, the Match transform redistributes the contribution allotted for this field to the other criteria and recalculates the contributions for the other fields.

Fields compared	Record A	Record B	% alike	Contribution	Score (per field)
Postcode	54601	54601	100	20 (or 22)	22
Address	100 Water St	100 Water St	100	40 (or 44)	44
Family_Name	Hamilton	Hammilton	94	30 (or 33)	31
Given_Name1	Mary		—	10 (or 0)	—
					Weighted score: 97

❖ Example

One field blank operation = Eval

One field blank score = 0

Input field: *Given_Name1*

Fields compared	Record A	Record B	% alike	Contribution	Score (per field)
Postcode	54601	54601	100	20	20
Address	100 Water St	100 Water St	100	40	40
Family_Name	Hamilton	Hammilton	94	30	28
Given_Name1	Mary		0	10	0
					Weighted score: 88

❖ Example

One field blank operation = Eval

One field blank score = 100

Input field: *Given_Name1*

Fields compared	Record A	Record B	% alike	Contribution	Score (per field)
Postcode	54601	54601	100	20	20
Address	100 Water St	100 Water St	100	40	40
Family_Name	Hamilton	Hammilton	94	30	28
Given_Name1	Mary		100	10	10
					Weighted score: 98

18.6.4.9.3.6 Multiple field (cross-field) comparisons

Multiple field comparisons enables you to match data in one field against data in multiple fields that may contain the same data.

In most cases, you compare data in one field against data in another field; a one to one ratio. For example, Field1 in the first record is compared with Field1 in the second record. However, there are situations when comparing one field to multiple fields can be useful; a one to many ratio. For example, you match values in a Phone field against values in fields such as Fax, Mobile, and Home.

Enable multiple field comparisons in the *Multiple Field Comparison* tab of a match criteria in the *Match Editor*. When you enable multiple field comparison, you choose to match selected fields against either all of the selected fields in each record, or against only the same field in each record.

i Note

By default, Match performs multiple field comparisons on match standard type fields. For example, Match automatically compares *Person1_Given_Name1* to *Person1_Given_Name_Match_Std1*. In this situation, you

do not explicitly enable multiple field comparison. In other words, no additional configuration is required to perform multiple field comparison against match standard fields.

18.6.4.9.3.6.1 How Match compares selected fields to all selected fields

When the transform compares each selected field to all selected fields in other records, it compares all fields that you define in the match criteria to each other.

→ Remember

“Selected” fields include the criteria field and the other fields you define in the [Additional fields to compare](#) table.

The transform compares selected fields to all selected fields in the following ways:

- If one or more field comparisons meet the settings for [Match score](#), the transform considers the two rows being compared as matches.
- If one or more field comparisons exceed the [No match score](#), the transform considers other defined criteria or weighted scoring to determine if the two rows are matches.

🔗 Example

You have the following input data in the Firm1 and Firm2 fields:

Row ID	Firm1	Firm2
1	Firstlogic	Postalsoft
2	SAP	Firstlogic

With the [Match score](#) set to 100 and [No match score](#) set to 99, the transform determines the two records as matching. The transform performs the following comparison process to come up with the match decision:

- First, the transform compares Row 1, Firm1 (Firstlogic) to Row 2, Firm1 (SAP). Without multi-field comparison enabled, the transform would make a No Match decision. However, with multi-field comparison, it doesn't make the No Match decision yet.
- Next, the transform compares Row 1, Firm2 to Row 2, Firm2. It continues with comparisons until it has made all other comparisons between all fields in all rows. Because Row 1, Firm1 (Firstlogic) and Row 2, Firm2 (Firstlogic) are 100% similar, the transform considers the two records as matches.

18.6.4.9.3.6.2 How Match compares selected fields to the same fields in other records

When the transform compares selected fields to the same fields in other records, it considers two records as matching when one of the designated fields matches .

When you set the Match transform to compare the same fields between records, you essentially set an OR condition within a match criteria. The transform uses each field to determine a match in the following way: If

Field_1, or Field_2, or Field_3 passes the match criteria, the records match. The *No Match Score* for one field does not automatically fail the criteria when you use multi-field comparison.

→ Remember

“Selected” fields include the criteria field and the other fields you define for the criteria.

♣ Example

Your input data contains a phone, fax, and cell phone field. If the data in any one of these input fields is the same between the rows, the transform determines the records to be matches.

Row ID	Phone	Fax	Cell
1	608-555-1234	608-555-0000	608-555-4321
2	608-555-4321	608-555-0000	608-555-1111

If you define the Phone and Cell fields in different match criteria, they would fail the following match criteria settings:

Match Score = 100

No Match Score = 99

However, because the match criteria is set up for comparing the selected fields to all selected fields in other records, and the Fax field values match exactly, the transform determines the Row 1 and Row 2 to be matches.

i Note

In the example above, the Cell field in Row 1 and the Phone field in Row 2 would not be considered a match with the selection of the *the same field to other records* option because it only compares within the same field in this case. If this cross-comparison is needed, select the *all selected fields in other records* option instead.

18.6.4.9.3.7 Proximity matching

Proximity matching gives you the ability to match records based on their proximity instead of comparing the string representation of data. You can match on geographic, numeric, and date proximity.

Related Information

[Matching on Geographic proximity \[page 456\]](#)

[Matching on numeric or date proximity \[page 457\]](#)

18.6.4.9.3.7.1 Matching on Geographic proximity

Geographic Proximity finds duplicate records based on geographic proximity, using latitude and longitude information. This is not driving distance, but Geographic distance. This option uses WGS 84 (GPS) coordinates.

The Geographic proximity option can:

- Search on objects within a radial range. This can help a company that wants to send a mailing out to customers within a certain distance from their business location.
- Search on the nearest location. This can help a consumer find a store location closest to their address.

18.6.4.9.3.7.2 Setting up Geographic Proximity matching - criteria fields

To select the fields for Geographic Proximity matching, follow these steps:

1. Access the Match Editor, add a new criteria.
2. From Available Criteria, expand [Geographic](#).
3. Select [LATITUDE_LONGITUDE](#).
This will make the two criteria fields available for mapping.
4. Map the correct latitude and longitude fields. You must map both fields.

18.6.4.9.3.7.3 Setting up Geographic Proximity matching - criteria options

You must have the Latitude and Longitude fields mapped before you can use this option.

To perform geographic proximity matching, follow these steps:

1. From Compare data using, select [Geo Proximity](#).
This filters the options under Comparison Rules to show only applicable options.
2. Set the Distance unit option to one of the following:
 - [Miles](#)
 - [Feet](#)
 - [Kilometers](#)
 - [Meters](#)
3. Enter the [Max Distance](#) you want to consider for the range.
4. Set the [Max Distance Score](#).

i Note

A distance equal to Max distance will receive a score of Max distance score. Any distance less than the Max distance will receive a proportional score between Max distance score and 100. For example, a proximity of 10 miles will have higher score than a 40 miles.

i Note

If the data for Max Distance may change from row to row, you should dynamically input the data using the Option_Field_Algorithm_Geo_Proximity_<logical_name>_Max_Distance field.

Related Information

[Reference Guide: Dynamic transform settings](#)

[Reference Guide: Transforms, Data Quality, Match, Input fields](#)

18.6.4.9.3.7.4 Matching on numeric or date proximity

The Match Transform's numeric proximity options find duplicates based on numerical closeness of data. You can find duplicates based on numeric values and date values. The following options are available in the Match Criteria Editor Options tab for numeric and date matching:

Numeric difference

Finds duplicates based on the numeric difference for numeric or date values. For example, you can use this option to find duplicates based on date values in a specific range (for example, plus or minus 35 days), regardless of character-based similarity.

Numeric percent difference

Finds duplicates based on the percentage of numeric difference for numeric values. Here are two examples where this might be useful :

- Finance data domain : You can search financial data to find all monthly mortgage payments that are within 5 percent of a given value.
- Product data domain, you can search product data to find all the steel rods that are within 10% tolerance of a given diameter.

18.6.4.10 Post-match processing

18.6.4.10.1 Best record

A key component in most data consolidation efforts is salvaging data from matching records—that is, members of match groups—and posting that data to a best record, or to all matching records.

You can perform these functions by adding a Best Record post-match operation.

Operations happen within match groups

The functions you perform with the Best Record operation involve manipulating or moving data contained in the master records and subordinate records of match groups. Match groups are groups of records that the Match transform has found to be matching, based on the criteria you have created.

A master record is the first record in the Match group. You can control which record this is by using a Group Prioritization operation before the Best Record operation.

Subordinate records are all of the remaining records in a match group.

To help illustrate this use of master and subordinate records, consider the following match group:

Record	Name	Phone	Date	Group rank
#1	John Smith		11 Apr 2001	Master
#2	John Smyth	788-8700	12 Oct 1999	Subordinate
#3	John E. Smith	788-1234	22 Feb 1997	Subordinate
#4	J. Smith	788-3271		Subordinate

Because this is a match group, all of the records are considered matching. As you can see, each record is slightly different. Some records have blank fields, some have a newer date, all have different phone numbers.

A common operation that you can perform in this match group is to move updated data to all of the records in a match group. You can choose to move data to the master record, to all the subordinate members of the match group, or to all members of the match group. The most recent phone number would be a good example here.

Another example might be to salvage useful data from matching records before discarding them. For example, when you run a drivers license file against your house file, you might pick up gender or date-of-birth data to add to your house record.

Post higher priority records first

The operations you set up in the Best Record option group should always start with the highest priority member of the match group (the master) and work their way down to the last subordinate, one at a time. This ensures that data can be salvaged from the higher-priority record to the lower priority record.

So, be sure that your records are prioritized correctly, by adding a Group Prioritization post-match operation before your Best Record operation.

18.6.4.10.1.1 Best record strategies

We provide you with strategies that help you set up some more common best record operation quickly and easily. If none of these strategies fit your needs, you can create a custom best record strategy, using your own Python code.

Best record strategies act as a criteria for taking action on other fields. If the criteria is not met, no action is taken.

Example

In our example of updating a phone field with the most recent data, we can use the Date strategy with the Newest priority to update the master record with the latest phone number in the match group. This latter part (updating the master record with the latest phone number) is the action. You can also update all of the records in the match group (master and all subordinates) or only the subordinates.

! Restriction

The date strategy does not parse the date, because it does not know how the data is formatted. Be sure your data is pre-formatted as YYYYMMDD, so that string comparisons work correctly. You can also do this by setting up a custom strategy, using Python code to parse the date and use a date compare.

18.6.4.10.1.1.1 Custom best record strategies and Python

In the pre-defined strategies for the Best Record strategies, the Match transform auto-generates the Python code that it uses for processing. Included in this code, are variables that are necessary to manage the processing.

Common variables

The common variables you see in the generated Python code are:

Variable	Description
SRC	Signifies the source field.
DST	Signifies the destination field.
RET	Specifies the return value, indicating whether the strategy passed or failed (must be either "T" or "F").

NEWDST and NEWGRP variables

Use the NEWDST and NEWGRP variables to allow the posting of data in your best-record action to be independent of the strategy fields. If you do not include these variables, the strategy field data must also be updated.

Variable	Description
NEWDST	New destination indicator. This string variable will have a value of "T" when the destination record is new or different than the last time the strategy was evaluated and a value of "F" when the destination record has not changed since last time. The NEWDST variable is only useful if you are posting to multiple destinations, such as ALL or SUBS in the Posting destination option.
NEWGRP	New group indicator. This string variable will have a value of "T" when the match group is different than the last time the strategy was evaluated and a value of "F" when the match group has not changed since last time.

NEWDST example

The following Python code was generated from a NON_BLANK strategy with options set this way:

Option	Setting
Best record strategy	NON_BLANK
Strategy priority	Priority option not available for the NON_BLANK strategy.
Strategy field	NORTH_AMERICAN_PHONE1_NORTH_AMERICAN_PHONE_STANDARDIZED.
Posting destination	ALL
Post only once per destination	YES

Here is what the Python code looks like.

```
# Setup local temp variable to store updated compare condition
dct = locals()
# Store source and destination values to temporary variables
```

```
# Reset the temporary variable when the destination changes
if (dct.has_key('BEST_RECORD_TEMP') and NEWDST.GetBuffer() == u'F'):
    DESTINATION = dct['BEST_RECORD_TEMP']
else:
    DESTINATION =
    DST.GetField(u'NORTH_AMERICAN_PHONE1 NORTH_AMERICAN_PHONE_STANDARDIZED')
    SOURCE = SRC.GetField(u'NORTH_AMERICAN_PHONE1 NORTH_AMERICAN_PHONE_STANDARDIZED')
    if len(SOURCE.strip()) > 0 and len(DESTINATION.strip()) == 0:
        RET.SetBuffer(u'T')
        dct['BEST_RECORD_TEMP'] = SOURCE
    else:
        RET.SetBuffer(u'F')
        dct['BEST_RECORD_TEMP'] = DESTINATION
# Delete temporary variables
del SOURCE
del DESTINATION
```

Example: NEWDST and NEWGRP

Suppose you have two match groups, each with three records.

Match group	Records
Match group 1	Record A
	Record B
	Record C
Match group 2	Record D
	Record E
	Record F

Each new destination or match group is flagged with a "T".

NEWGRP (T or F)	NEWDST (T or F)	Comparison
T (New match group)	T (New destination "A")	Record A > Record B
F	F	A > C
F	T (New destination "B")	B > A
F	F	B > C
F	T (New destination "C")	C > A
F	F	C > B
T (New match group)	T (New destination "D")	D > E
F	F	D > F
F	T (New destination "E")	E > D
F	F	E > F
F	T (New destination "F")	F > D

NEWGRP (T or F)	NEWDST (T or F)	Comparison
F	F	F > E

18.6.4.10.1.1.2 Creating a pre-defined best record strategy

Be sure to add a Best Record post-match operation to the appropriate match level in the Match Editor. Also, remember to map any pertinent input fields to make them available for this operation.

This procedure allows you to quickly generate the criteria for your best record action. The available strategies reflect common use cases.

1. Enter a name for this Best Record operation.
2. Select a strategy from the [Best record strategy](#) option.
3. Select a priority from the [Strategy priority](#) option.
The selection of values depends on the strategy you chose in the previous step.
4. Select a field from the [Strategy field](#) drop-down menu.
The field you select here is the one that acts as a criteria for determining whether a best record action is taken.

Example

The strategy field you choose must contain data that matches the strategy you are creating. For example, if you are using a newest date strategy, be sure that the field you choose contains date data.

18.6.4.10.1.1.3 Creating a custom best record strategy

1. Add a best record operation to your Match transform.
2. Enter a name for your best record operation.
3. In the [Best record strategy](#) option, choose Custom.
4. Choose a field from the [Strategy field](#) drop-down list.
5. Click the [View/Edit Python](#) button to create your custom Python code to reflect your custom strategy.
The Python Editor window appears.

18.6.4.10.1.2 Best record actions

Best record actions are the functions you perform on data if a criteria of a strategy is met.

Example

Suppose you want to update phone numbers of the master record. You would only want to do this if there is a subordinate record in the match group that has a newer date, which signifies a potentially new phone number for that person.

The action you set up would tell the Match transform to update the phone number field in the master record (action) if a newer date in the date field is found (strategy).

18.6.4.10.1.2.1 Sources and destinations

When working with the best record operation, it is important to know the differences between sources and destinations in a best record action.

The source is the field from which you take data and the destination is where you post the data. A source or destination can be either a master or subordinate record in a match group.

Example

In our phone number example, the subordinate record has the newer date, so we take data from the phone field (the source) and post it to the master record (the destination).

18.6.4.10.1.2.2 Posting once or many times per destination

In the Best Record options, you can choose to post to a destination once or many times per action by setting the *Post only once per destination* option.

You may want your best record action to stop after the first time it posts data to the destination record, or you may want it to continue with the other match group records as well. Your choice depends on the nature of the data you're posting and the records you're posting to. The two examples that follow illustrate each case.

If you post only once to each destination record, then once data is posted for a particular record, the Match transform moves on to either perform the next best record action (if more than one is defined) or to the next record.

If you don't limit the action in this way, all actions are performed each time the strategy returns True.

Regardless of this setting, the Match transform always works through the match group members in priority order. When posting to record #1 in the figure below, without limiting the posting to only once, here is what happens:

Match group	Action
Record #1 (master)	

Match group	Action
Record #2 (subordinate)	First, the action is attempted using, as a source, that record from among the other match group records that has the highest priority (record #2).
Record #3 (subordinate)	Next, the action is attempted with the next highest priority record (record #3) as the source.
Record #4 (subordinate)	Finally, the action is attempted with the lowest priority record (record #4) as the source.

The results In the case above, record #4 was the last source for the action, and therefore could be a source of data for the output record. However, if you set your best record action to post only once per destination record, here is what happens:

Match group	Action
Record #1 (master)	
Record #2 (subordinate)	First, the action is attempted using, as a source, that record from among the other match group records that has the highest priority (record #2). If this attempt is successful, the Match transform considers this best record action to be complete and moves to the next best record action (if there is one), or to the next output record. If this attempt is not successful, the Match transform moves to the match group member with the next highest priority and attempts the posting operation.
Record #3 (subordinate)	
Record #4 (subordinate)	

In this case, record #2 was the source last used for the best record action, and so is the source of posted data in the output record.

18.6.4.10.1.2.3 Creating a best record action

The best record action is the posting of data from a source to a destination record, based on the criteria of your best record strategy.

1. Create a strategy, either pre-defined or custom.
2. Select the record(s) to post to in the *Posting destination* option.
3. Select whether you want to post only once or multiple times to a destination record in the *Post only once per destination* option.
4. In the *Best record action fields table*, choose your source field and destination field.
When you choose a source field, the *Destination field* column is automatically populated with the same field. You need to change the destination field if this is not the field you want to post your data to.
5. If you want to create a custom best record action, choose Yes in the *Custom* column.
You can now access the Python editor to create custom Python code for your custom action.

18.6.4.10.1.3 Destination protection

The Best Record and Unique ID operations in the Match transform offer you the power to modify existing records in your data. There may be times when you would like to protect data in particular records or data in records from particular input sources from being overwritten.

The Destination Protection tab in these Match transform operations allow you the ability to protect data from being modified.

18.6.4.10.1.3.1 Protecting destination records through fields

1. In the Destination Protection tab, select *Enable destination protection*.
2. Select a value in the *Default destination protection* option drop-down list.
This value determines whether a destination is protected if the destination protection field does not have a valid value.
3. Select the *Specify destination protection by field* option, and choose a field from the *Destination protection field* drop-down list (or *Unique ID protected field*).
The field you choose must have a Y or N value to specify the action.

Any record that has a value of Y in the destination protection field will be protected from being modified.

18.6.4.10.1.3.2 Protecting destination records based on input source membership

You must add an Input Source operation and define input sources before you can complete this task.

1. In the Destination Protection tab, select *Enable destination protection*.
2. Select a value in the *Default destination protection* option drop-down list.
This value determines whether a destination (input source) is protected if you do not specifically define the source in the table.
3. Select the *Specify destination protection by source* option.
4. Select an input source from the first row of the *Source name* column, and then choose a value from the *Destination protected* (or *Unique ID protected*) column.
Repeat for every input source you want to set protection for. Remember that if you do not specify for every source, the default value will be used.

18.6.4.10.2 Unique ID

A unique ID refers to a field within your data which contains a unique value that is associated with a record or group of records. You could use a unique ID, for example, in your company's internal database that receives updates at some predetermined interval, such as each week, month, or quarter. Unique ID applies to a data

record in the same way that a national identification number might apply to a person; for example, a Social Security number (SSN) in the United States, or a National Insurance number (NINO) in the United Kingdom. It creates and tracks data relationships from run to run. With the Unique ID operation, you can set your own starting ID for new key generation, or have it dynamically assigned based on existing data. The Unique ID post-match processing operation also lets you begin where the highest unique ID from the previous run ended.

Unique ID works on match groups

Unique ID doesn't necessarily assign IDs to individual records. It can assign the same ID to every record in a match group (groups of records found to be matches).

If you are assigning IDs directly to a break group, use the *Group number field* option to indicate which records belong together. Additionally, make sure that the records are sorted by group number so that records with the same group number value appear together.

If you are assigning IDs to records that belong to a match group resulting from the matching process, the *Group number field* is not required and should not be used.

i Note

If you are assigning IDs directly to a break group and the *Group number field* is not specified, Match treats the entire data collection as one match group.

18.6.4.10.2.1 Unique ID processing options

The Unique ID post-match processing operation combines the update source information with the master database information to form one source of match group information. The operation can then assign, combine, split, and delete unique IDs as needed. You can accomplish this by using the *Processing operation* option.

Operation	Description
Assign	Assigns a new ID to unique records that don't have an ID or to all members of a group that don't have an ID. In addition, the assign operation copies an existing ID if a member of a match group already has an ID. Each record is assigned a value. - Records in a match group where one record had an input unique ID will share the value with other records in the match group which had no input value. The first value encountered will be shared. Order affects this; if you have a priority field that can be sequenced using ascending order, place a Prioritization post-match operation prior to the Unique ID operation. - Records in a match group where two or more records had different unique ID input values will each keep their input value. - If all of the records in a match group do not have an input unique ID value, then the next available ID will be assigned to each record in the match group. If the GROUP_NUMBER input field is used, then records with the same group number must appear consecutively in the data collection. i Note Use the GROUP_NUMBER input field only when processing a break group that may contain smaller match groups. If the GROUP_NUMBER field is not specified, Unique ID assumes that the entire collection is one group.
AssignCombine	Performs both an Assign and a Combine operation. Each record is assigned a value. - Records that did not have an input unique ID value and are not found to match another record containing an input unique ID value will have the next available ID assigned to it. These are "add" records that could be unique records or could be matches, but not to another record that had previously been assigned a unique ID value. - Records in a match group where one or more records had an input unique ID with the same or different values will share the first value encountered with all other records in the match group. Order affects this; if you have a priority field that can be sequenced using ascending order, place a Prioritization post-match operation prior to the Unique ID operation. If the GROUP_NUMBER input field is used, then records with the same group number must appear consecutively in the data collection. i Note Use the GROUP_NUMBER input field only when processing a break group that may contain smaller match groups. If the GROUP_NUMBER field is not specified, Unique ID assumes that the entire collection is one group.

Operation	Description
Combine	Ensures that records in the same match group have the same Unique ID. For example, this operation could be used to assign all the members of a household the same unique ID. Specifically, if a household has two members that share a common unique ID, and a third person moves in with a different unique ID, then the Combine operation could be used to assign the same ID to all three members. The first record in a match group that has a unique ID is the record with the highest priority. All other records in the match group are given this record's ID (assuming the record is not protected). The Combine operation does not assign a unique ID to any record that does not already have a unique ID. It only combines the unique ID of records in a match group that already have a unique ID. If the GROUP_NUMBER input field is used, then records with the same group number must appear consecutively in the data collection. i Note Use the GROUP_NUMBER input field only when processing a break group that may contain smaller match groups. If the GROUP_NUMBER field is not specified, Unique ID assumes that the entire collection is one group.
Delete	Deletes unique IDs from records that no longer need them, provided that they are not protected from being deleted. If you are using a file and are recycling IDs, this ID is added to the file. When performing a delete, records with the same unique ID should be grouped together. When Match detects that a group of records with the same unique ID is about to be deleted: • If any of the records are protected, all records in the group are assumed to be protected. • If recycling is enabled, the unique ID will be recycled only once, even though a group of records had the same ID.
Split	Changes a split group's unique records, so that the records that do not belong to the same match group will have a different ID. The record with the group's highest priority will keep its unique ID. The rest will be assigned new unique IDs. For this operation, you must group your records by unique ID, rather than by match group number. For example: • Records in a match group where two or more records had different unique ID input values or blank values will each retain their input value, filled or blank depending on the record. • Records that did not have an input unique ID value and did not match any record with an input unique ID value will have a blank unique ID on output. • Records that came in with the same input unique ID value that no longer are found as matches have the first record output with the input value. Subsequent records are assigned new unique ID values.

18.6.4.10.2.2 Unique ID protection

The output for the unique ID depends on whether an input field in that record has a value that indicates that the ID is protected.

- If the protected unique ID field is not mapped as an input field, Match assumes that none of the records are protected.
- There are two valid values allowed in this field: Y and N. Any other value is converted to Y.
A value of N means that the unique ID is not protected and the ID posted on output may be different from the input ID.
a value of Y means that the unique ID is protected and the ID posted on output will be the same as the input ID.
- If the protected unique ID field is mapped as an input field, a value other than N means that the record's input data will be retained in the output unique ID field.

These rules for protected fields apply to all unique ID processing operations.

18.6.4.10.2.3 Unique ID limitations

Because some options in the unique ID operation are based on reading a file or referring to a field value, there may be implications for when you are running a multi-server or real-time server environment and sharing a unique ID file.

- If you are reading from or writing to a file, the unique ID file must be on a shared file system.
- Recycled IDs are used in first-in, first-out order. When Match recycles an ID, it does not check whether the ID is already present in the file. You must ensure that a particular unique ID value is not recycled more than once.

18.6.4.10.2.4 Assigning unique IDs using a file

1. In the Unique ID option group, select the *Value from file* option.
2. Set the file name and path in the *File* option.

This file must be an XML file and must adhere to the following structure:

```
<UniqueIdSession>
  <CurrentUniqueId>477</CurrentUniqueId>
</UniqueIdSession>
```

i Note

The value of 477 is an example of a starting value. However, the value must be 1 or greater.

18.6.4.10.2.5 Assigning a unique ID using a constant

Similar to using a file, you can assign a starting unique ID by defining that value.

1. Select the *Constant value* option.
2. Set the *Starting value* option to the desired ID value.

18.6.4.10.2.6 Assigning unique IDs using a field

The Field option allows you to send the starting unique ID through a field in your data source or from a User-Defined transform, for example.

The starting unique ID is passed to the Match transform before the first new unique ID is requested. If no unique ID is received, the starting number will default to 1.

⚠ Caution

Use caution when using the Field option. The field that you use must contain the unique ID value you want to begin the sequential numbering with. This means that each record you process must contain this field, and each record must have the same value in this field.

For example, suppose the value you use is 100,000. During processing, the first record or match group will have an ID of 100,001. The second record or match group receives an ID of 100,002, and so on.

The value in the first record that makes it to the Match transform contains the value where the incrementing begins.

There is no way to predict which record will make it to the Match transform first (due to sorting, for example); therefore, you cannot be sure which value the incrementing will begin at.

1. Select the *Field* option.
2. In the *Starting unique ID field* option, select the field that contains the starting unique ID value.

18.6.4.10.2.7 Assigning unique IDs using GUID

You can use Globally Unique Identifiers (GUID) as unique IDs.

Select the *GUID* option.

GUID is also known as the Universal Unique Identifier (UUID). The UUID variation used for unique ID is a time-based 36-character string with the format: TimeLow-TimeMid-TimeHighAndVersion-ClockSeqAndReservedClockSeqLow-Node

For more information about UUID, see the Request for Comments (RFC) document at <http://www.ietf.org/rfc/rfc4122.txt> .

18.6.4.10.2.8 Recycling unique IDs

If unique IDs are dropped during the Delete processing option, you can write those IDs back to a file to be used later.

1. In the Unique ID option group, set the *Processing operation* option to *Delete*.
2. Select the *Value from file* option.
3. Set the file name and path in the *File* option.
4. Set the *Recycle unique IDs* option to *Yes*. This is the same file that you might use for assigning a beginning ID number.

18.6.4.10.2.8.1 Using your own recycled unique IDs

If you have some IDs of your own that you would like to recycle and use in a data flow, you can enter them in the file you want to use for recycling IDs and posting a starting value for your IDs. Enter these IDs in an XML tag of `<R></R>`. For example:

```
<UniqueIdSession>
  <CurrentUniqueId>477</CurrentUniqueId>
  <R>214</R>
  <R>378</R>
</UniqueIdSession>
```

18.6.4.10.2.9 Destination protection

The Best Record and Unique ID operations in the Match transform offer you the power to modify existing records in your data. There may be times when you would like to protect data in particular records or data in records from particular input sources from being overwritten.

The Destination Protection tab in these Match transform operations allow you the ability to protect data from being modified.

18.6.4.10.2.9.1 Protecting destination records through fields

1. In the Destination Protection tab, select *Enable destination protection*.
2. Select a value in the *Default destination protection* option drop-down list.
This value determines whether a destination is protected if the destination protection field does not have a valid value.
3. Select the *Specify destination protection by field* option, and choose a field from the *Destination protection field* drop-down list (or *Unique ID protected field*).
The field you choose must have a Y or N value to specify the action.

Any record that has a value of Y in the destination protection field will be protected from being modified.

18.6.4.10.2.9.2 Protecting destination records based on input source membership

You must add an Input Source operation and define input sources before you can complete this task.

1. In the Destination Protection tab, select *Enable destination protection*.
2. Select a value in the *Default destination protection* option drop-down list.
This value determines whether a destination (input source) is protected if you do not specifically define the source in the table.
3. Select the *Specify destination protection by source* option.
4. Select an input source from the first row of the *Source name* column, and then choose a value from the *Destination protected* (or *Unique ID protected*) column.
Repeat for every input source you want to set protection for. Remember that if you do not specify for every source, the default value will be used.

18.6.4.10.3 Group statistics

The Group Statistics post-match operation should be added after any match level and any post-match operation for which you need statistics about your match groups or your input sources.

This operation can also counts statistics from logical input sources that you have already identified with values in a field (pre-defined) or from logical sources that you specify in the Input Sources operation.

This operation also allows you to exclude certain logical sources based on your criteria.

i Note

If you choose to count input source statistics in the Group Statistics operation, Match will also count basic statistics about your match groups.

Group statistics fields

When you include a Group Statistics operation in your Match transform, the following fields are generated by default:

- GROUP_COUNT
- GROUP_ORDER
- GROUP_RANK
- GROUP_TYPE

In addition, if you choose to generate source statistics, the following fields are also generated and available for output:

- SOURCE_COUNT
- SOURCE_ID

- SOURCE_ID_COUNT
- SOURCE_TYPE_ID

Related Information

[Reference Guide: Transforms, Match, Output fields](#)

[Management Console Guide: Data Quality Reports, Match Source Statistics Summary report](#)

18.6.4.10.3.1 Generating only basic statistics

This task will generate statistics about your match groups, such as how many records in each match group, which records are masters or subordinates, and so on.

1. Add a Group Statistics operation to each match level you want, by selecting *Post Match Processing* in a match level, clicking the *Add* button, and selecting *Group Statistics*.
2. Select *Generate only basic statistics*.
3. Click the *Apply* button to save your changes.

18.6.4.10.3.2 Generating statistics for all input sources

Before you start this task, be sure that you have defined your input sources in the Input Sources operation.

Use this procedure if you are interested in generating statistics for all of your sources in the job.

1. Add a Group Statistics operation to the appropriate match level.
2. Select the *Generate source statistics from input sources* option.

This will generate statistics for all of the input sources you defined in the Input Sources operation.

18.6.4.10.3.3 Counting statistics for input sources generated by values in a field

For this task, you do not need to define input sources with the Input Sources operation. You can specify input sources for Match using values in a field.

Using this task, you can generate statistics for all input sources identified through values in a field, or you can generate statistics for a sub-set of input sources.

1. Add a Group Statistics operation to the appropriate match level.
2. Select the *Generate source statistics from source values* option.
3. Select a field from the *Logical source field* drop-down list that contains the values for your logical sources.

4. Enter a value in the *Default logical source value* field.
This value is used if the logical source field is empty.
5. Select one of the following:

Option	Description
<i>Count all sources</i>	Select to count all sources. If you select this option, you can click the Apply button to save your changes. This task is complete.
<i>Choose sources to count</i>	Select to define a sub-set of input sources to count. If you select this option, you can proceed to step 6 in the task.

6. Choose the appropriate value in the *Default count flag* option.
Choose Yes to count any source not specified in the *Manually define logical source count flags table*. If you do not specify any sources in the table, you are, in effect, counting all sources.
7. Select *Auto-generate sources* to count sources based on a value in a field specified in the *Predefined count flag field* option.
If you do not specify any sources in the *Manually define logical source count flags table*, you are telling the Match transform to count all sources based on the (Yes or No) value in this field.
8. In the *Manually define logical source count flags table*, add as many rows as you need to include all of the sources you want to count.

i Note

This is the first thing the Match transform looks at when determining whether to count sources.

9. Add a source value and count flag to each row, to tell the Match transform which sources to count.

→ Tip

If you have a lot of sources, but you only want to count two, you could speed up your set up time by setting the *Default count flag* option to No, and setting up the *Manually define logical source count flags table* to count those two sources. Using the same method, you can set up Group Statistics to count everything and not count only a couple of sources.

18.6.4.10.4 Output flag selection

By adding an Output Flag Selection operation to each match level (Post Match Processing) you want, you can flag specific record types for evaluation or routing downstream in your data flow.

Adding this operation generates the Select_Record output field for you to include in your output schema. This output field is populated with a Y or N depending on the type of record you select in the operation.

Your results will appear in the Match Input Source Output Select report. In that report, you can determine which records came from which source or source group and how many of each type of record were output per source or source group.

Record type	Description
<i>Unique</i>	Records that are not members of any match group. No matching records were found. These can be from sources with a normal or special source.

Record type	Description
<i>Single source masters</i>	Highest ranking member of a match group whose members all came from the same source. Can be from normal or special sources.
<i>Single source subordinates</i>	A record that came from a normal or special source and is a subordinate member of a match group.
<i>Multiple source masters</i>	Highest ranking member of a match group whose members came from two or more sources. Can be from normal or special sources.
<i>Multiple source subordinates</i>	A subordinate record of a match group that came from a normal or special source whose members came from two or more sources.
<i>Suppression matches</i>	Subordinate member of a match group that includes a higher-priority record that came from a suppress-type source. Can be from normal or special source.
<i>Suppression uniques</i>	Records that came from a suppress source for which no matching records were found.
<i>Suppression masters</i>	A record that came from a suppress source and is the highest ranking member of a match group.
<i>Suppression subordinates</i>	A record that came from a suppress-type source and is a subordinate member of a match group.

18.6.4.10.4.1 Flagging source record types for possible output

1. In the Match editor, for each match level you want, add an Output Flag Select operation.
2. Select the types of records for which you want to populate the Select_Record field with Y.

The Select_Record output field can then be output from the Match transform for use downstream in the data flow. This is most helpful if you later want to split off suppression matches or suppression masters from your data (by using a Case transform, for example).

18.6.4.11 Association matching

Association matching combines the matching results of two or more match sets (transforms) to find matches that could not be found within a single match set.

You can set up association matching in the Associate transform. This transform acts as another match set in your data flow, from which you can derive statistics.

This match set has two purposes. First, it provides access to any of the generated data from all match levels of all match sets. Second, it provides the overlapped results of multiple criteria, such as name and address, with name and SSN, as a single ID. This is commonly referred to as association matching.

Group numbers

The Associate transform accepts a group number field, generated by the Match transforms, for each match result that will be combined. The transform can then output a new associated group number.

The Associate transform can operate either on all the input records or on one data collection at a time. The latter is needed for real-time support.

Example: Association example

Say you work at a technical college and you want to send information to all of the students prior to the start of a new school year. You know that many of the students have a temporary local address and a permanent home address.

In this example, you can match on name, address, and postal code in one match set, and match on name and Social Security number (SSN), which is available to the technical college on every student, in another match set.

Then, the Associate transform combines the two match sets to build associated match groups. This lets you identify people who may have multiple addresses, thereby maximizing your one-to-one marketing and mailing efforts.

18.6.4.12 Unicode matching

Unicode matching lets you match Unicode data. You can process any non-Latin1 Unicode data, with special processing for Chinese, Japanese, Korean and Taiwanese (or CJKT) data.

Chinese, Japanese, Korean, and Taiwanese matching

Regardless of the country-specific language, the matching process for CJKT data is the same. For example, the Match transform:

- Considers half-width and full-width characters to be equal.
- Considers native script numerals and Arabic numerals to be equal. It can interpret numbers that are written in native script. This can be controlled with the Convert text to numbers option in the Criteria options group.
- Includes variations for popular, personal, and firm name characters in the referential data.
- Considers firm words, such as Corporation or Limited, to be equal to their variations (Corp. or Ltd.) during the matching comparison process. To find the abbreviations, the transform uses native script variations of the English alphabets during firm name matching.
- Ignores commonly used optional markers for province, city, district, and so on, in address data comparison.
- Intelligently handles variations in a building marker.

Japanese-specific matching capabilities

With Japanese data, the Match transform considers:

- Block data markers, such as chome and banchi, to be equal to those used with hyphenated data.
- Words with or without Okurigana to be equal in address data.
- Variations of *no* marker, *ga* marker, and so on, to be equal.
- Variations of a hyphen or dashed line to be equal.

Unicode match limitations

The Unicode match functionality does not:

- Perform conversions of simplified and traditional Chinese data.
- Match between non-phonetic scripts like kanji, simplified Chinese, and so on.

Route records based on country ID before matching

Before sending Unicode data into the matching process, you must first, as best you can, separate out the data by country to separate match transforms. This can be done by using a Case transform to route country data based on the country ID.

→ Tip

The Match wizard can do this for you when you use the multi-national strategy.

Inter-script matching

Inter-script matching allows you to process data that may contain more than one script by converting the scripts to Latin1. For example one record has Latin1 and other has katakana, or one has Latin and other has Cyrillic. Select [Yes](#) to enable Inter-script matching. If you prefer to process the data without converting it to Latin1, leave the Inter-script Matching option set [No](#). Here are two examples of names matched using inter-script matching:

Name	Can be matched to...
Viktor Ivanov	Виктор Иванов
Takeda Noburu	スッセフレ

Locale

The Locale option specifies the locale setting for the criteria field. Setting this option is recommended if you plan to use the Text to Numbers feature to specify the locale of the data for locale-specific text-to-number conversion for the purpose of matching. Here are four examples of text-to-number conversion:

Language	Text	Numbers
French	quatre mille cinq cents soixante-sept	4567
German	dreitausendzwei	3002
Italian	cento	100
Spanish	ciento veintisiete	127

For more information on these matching options, see the Match Transform section of the *Reference Guide*.

18.6.4.12.1 Setting up Unicode matching

1. Use a Case transform to route your data to a Match transform that handles that type of data.
2. Open the [AddressJapan_MatchBatch](#) Match transform configuration, and save it with a different name.
3. Set the [Match engine](#) option in the Match transform options to a value that reflects the type of data being processed.
4. Set up your criteria and other desired operations. For more information on Match Criteria options, see the Match Transform section of the *Reference Guide*.

Example

- When possible, use criteria for parsed components for address, firm, and name data, such as Primary_Name or Person1_Family_Name1.
- If you have parsed address, firm, or name data that does not have a corresponding criteria, use the Address_Data1-5, Firm_Data1-3, and Name_Data1-3 criteria.
- For all other data that does not have a corresponding criteria, use the Custom criteria.

18.6.4.13 Phonetic match criteria

There are instances where using phonetic data can produce more matches when used as a criteria, than if you were to match on other criteria such as name or firm data.

Use the Double Metaphone or Soundex functions to populate a field and use it for creating break groups or use it as a criteria in matching.

Function	Description
Double metaphone	Encodes the input string using the Double Metaphone algorithm and returns a string.
Soundex	Encodes the input string using the Soundex algorithm and returns a string. Use when you want to push down the function to the database-level.

Matching on name field data produces different results than matching on phonetic data.

Example

Name	Comparison score
Smith	72% similar
Smythe	

Name	Phonetic key (primary)	Comparison score
Smith	SMO	100% similar
Smythe	SMO	

Related Information

Reference Guide, Transforms, Match, Matching, Match criteria options: Options tab

[Reference Guide, Functions and procedures, Descriptions of built-in functions \(double_metaphone and soundex\)](#)

18.6.4.13.1 Phonetic matching criteria options

If you intend to match on phonetic data, set up criteria options this way.

Option	Value
Compare data using	Field Similarity compares the entire field's data as a single string.
Check for transposed letters	Set to No. Software treats transposed characters the same way it handles any non-matching characters.
Initials adjustment score	Set to 0 (zero) (default) to disable initials checking.

Option	Value
Substring adjustment score	Set to 0 (zero) to disable substring checking (default).
Abbreviation adjustment score	Set to 0 (zero) to disable abbreviation checking.

Related Information

Reference Guide, Transforms, Match, Matching, Match criteria options: Options tab

18.6.4.13.2 Phonectic matching match scores

If you are matching only on the phonetic criteria, set your match score options like this.

Option	Value
Match score	Set to 100 to specify the minimum similarity score needed for the records to be considered a match based on this criteria.
No match score	Set to 99 to specify the maximum similarity score needed for the records to be considered a no-match based on this criteria.

If you are matching on multiple criteria, including a phonetic criteria, place the phonetic criteria first in the order of criteria and set your match score options like this:

Option	Value
Match score	Set to 101 to ensure that the software doesn't use just this criteria to consider two records a match and that it needs consider other criteria in the comparison process.
No match score	Set to 99 to specify the maximum similarity score needed for the records to be considered a no-match based on this criteria.

Related Information

Reference Guide, Transforms, Match, Matching, Match criteria options: Options tab

18.6.4.13.3 Phonetic matching blank fields

When you use break groups, records that have no value are not in the same group as records that have a value (unless you set up matching on blank fields).

Example

Consider the following two input records:

Mr Johnson	100 Main St	La Crosse	WI	54601
Scott Johnson	100 Main St	La Crosse	WI	54601

After these records are processed by the Data Cleanse transform, the first record will have an empty first name field and, therefore, an empty phonetic field. This means that there cannot be a match if you are creating break groups. If you are not creating break groups, there cannot be a match if you are not blank-matching.

18.6.4.13.4 Phonectic matching length of data

The length you assign to a phonetic function output is important as this example shows.

Example

First name (last name)	Output
S (Johnson)	S
Scott (Johnson)	SKT

Suppose these two records represent the same person. In this example, if you break on more than one character, these records will be in different break groups and, therefore, will not be compared.

18.6.4.14 Set up for match reports

Generate match reports to help analyze match results.

We offer many match reports to help you analyze your match results:

- Match Contribution report

- Match Criteria Summary
- Match Duplicate Sample report
- Match Input Source Output Select report
- Match Multi-source Frequency report
- Match Source Statistics Summary report

Related Information

[Management Console Guide: Data Quality Reports](#)

18.6.4.14.1 Include Group Statistics in your Match transform

To generate the Match Source Statistics Summary report include a Group Statistics operation in your Match and Associate transform(s).

If you want to track your input source statistics, you may want to include an Input Sources operation in the Match transform to define your sources and, in a Group Statistics operation select to generate statistics for your input sources.

i Note

You can also generate input source statistics in the Group Statistics operation by defining input sources using field values. You do not necessarily need to include an Input Sources operation in the Match transform.

18.6.4.14.2 Generate report statistics

To generate the data you want to see in match reports (other than the Match Source Statistics report), you must set the [Generate report statistics](#) option to Yes in the Match and Associate transform(s).

By turning on report data generation, you can get information about break groups, which criteria were instrumental in creating a match, and so on.

i Note

Be aware that turning on the report option can have an impact on your processing performance. It's best to turn on report data generation after you have thoroughly tested your data flow.

18.6.4.14.3 Use unique names for match sets, levels, and operations

Use unique names in the Match and Associate transforms for your match sets, levels, and each of your pre- and post-match operations

Use unique names in the Match and Associate transforms for your match sets, levels, and each of your pre- and post-match operations. Using unique names helps you get the most accurate data in your reports such as Group Prioritization and Group Statistics, and it helps you better understand which of these elements is producing the data you are looking at.

18.6.4.14.4 Insert appropriate output fields

Include these three output fields in the Match transform to post data into the Match Duplicate Sample report

Field	Description
Match_Type	Indicates the type of match that brought the record into a match group: • blank • D = driver • R = rule • W = weighted
Group_Number	Specifies the records that belong to the same match group and share the same group number.
Match_Score	Outputs the following information: • Criteria or pattern similarity score • Total weighted score • Blank

Related Information

[Reference Guide, Transforms, Data Quality transforms, Match, Output fields](#)

18.6.5 Address Cleanse

Address cleanse provides a corrected, complete, and standardized form of your original address data.

SAP Data Services enables you to create applications that use many address cleanse features such as basic parsing and standardizing and more advanced concepts that are unique to specific transforms:

[How address cleanse works \[page 484\]](#)

Cleanse your address data using both the USA Regulatory Address Cleanse transform and the Global Address Cleanse transform.

[Address cleanse reports \[page 489\]](#)

Both USA Regulatory Address Cleanse and Global Address Cleanse transforms enable you to generate certified postal reports.

[Preparing your input data \[page 490\]](#)

Before you start with address cleansing, decide what kind of address line format to input.

[Determining which transform\(s\) to use \[page 492\]](#)

[Identifying the country of destination \[page 497\]](#)

[Set up the reference files \[page 498\]](#)

The USA Regulatory Address Cleanse transform and the Global Address Cleanse transform and engines rely on several directories (reference files) to cleanse your data.

[Defining the standardization options \[page 501\]](#)

[Process Japanese addresses \[page 503\]](#)

The Global Address Cleanse transform and the Global Address engine parses and normalizes Japanese addresses for address cleansing and data matching.

[Process Chinese addresses \[page 514\]](#)

The Global Address engine in the Global Address Cleanse transform parses Chinese addresses.

[Supported countries \(Global Address Cleanse\) \[page 519\]](#)

[New Zealand certification \[page 522\]](#)

[Global Address Cleanse Suggestion List option \[page 527\]](#)

The Global Address Cleanse transform's Suggestion List processing option is used in transactional projects to complete and populate addresses that have minimal data.

[Global Suggestion List transform \[page 528\]](#)

[Address assignment for buildings \[page 530\]](#)

The Global Address Cleanse transform uses building information in addresses to enhance the address assignment process and results.

18.6.5.1 How address cleanse works

Cleanse your address data using both the USA Regulatory Address Cleanse transform and the Global Address Cleanse transform.

The following table describes how the two transforms cleanse your data.

Process	Description
Verifies code consistency	Uses postal directories to verify that a postal code matches the locality and region. Adds missing postal elements when other elements are present and verified. For example, if your data has a locality and region, adds the corresponding postal code depending on the country.

Process	Description
Standardizes address lines	Uses the address standardization options that you set in the transform to output addresses that include or exclude all punctuation, uses all upper or all lower case text, or abbreviates or spells out address types, for example.
Identifies undeliverable addresses	For USA records only, identifies when an address is undeliverable, such as a vacant lot or a condemned building.
Assigns diagnostic codes	Assigns established codes that indicate reasons for unsigned addresses, or how addresses were corrected.

[Address Cleanse transforms \[page 485\]](#)

The address cleanse transforms provide different cleansing functionalities.

[Input and output data and field classes \[page 488\]](#)

Use address cleanse input and output fields based on the field format and the type of data the field contains.

Parent topic: [Address Cleanse \[page 483\]](#)

Related Information

[Address cleanse reports \[page 489\]](#)

[Preparing your input data \[page 490\]](#)

[Determining which transform\(s\) to use \[page 492\]](#)

[Identifying the country of destination \[page 497\]](#)

[Set up the reference files \[page 498\]](#)

[Defining the standardization options \[page 501\]](#)

[Process Japanese addresses \[page 503\]](#)

[Process Chinese addresses \[page 514\]](#)

[Supported countries \(Global Address Cleanse\) \[page 519\]](#)

[New Zealand certification \[page 522\]](#)

[Global Address Cleanse Suggestion List option \[page 527\]](#)

[Global Suggestion List transform \[page 528\]](#)

[Address assignment for buildings \[page 530\]](#)

18.6.5.1.1 Address Cleanse transforms

The address cleanse transforms provide different cleansing functionalities.

The following table describes the transforms that provide address cleanse processing for your data.

Transform	Description
DSF2 Walk Sequencer	Adds delivery sequence information to your data when you perform DSF2 walk sequencing, which you can use with pre-sorting software to qualify for walk-sequence discounts. i Note The software does not place your data in walk sequence order.
Global Address Cleanse and engines	Cleanses your address data from any of the supported countries, excluding U.S. certification. Set up the Global Address Cleanse transform in conjunction with one or more of the Global Address Cleanse engines: Canada, Global Address, or USA. Create mailings that comply with the following postal authority requirements: • Canada Post Software Evaluation and Recognition Program (SERP)—Statement of Address Accuracy Report • Australia Post Address Matching Processing Summary report (AMAS) • New Zealand Statement of Accuracy (SOA) report Additionally offers global suggestion lists and country ID options: • Global suggestion list offers suggestions for possible address matches for your USA, Canada, and global address data. • Country ID identifies the country of destination for the record and outputs an ISO code.

Transform	Description
USA Regulatory Address Cleanse	Identifies, parses, validates, and corrects USA address data, within the Latin 1 code page, according to the CASS (U.S. Coding Accuracy Support System). Generates the USPS Form 3553 and outputs many useful codes to your records. Choose to run in non-certified mode and utilize suggestion list functionality. Offers the following USPS postal service options: • DPV • DSF2 (augment) • eLOT • EWS • GeoCensus • LACSLink • NCOALink • RDI • SuiteLink • Z4Change
Global Suggestion Lists	Offers suggestions for possible address matches for your USA, Canada, and global address data. This transform is usually used for real-time processing and does not standardize addresses. Use a Country ID transform before this transform in the data flow. Also, if you want to standardize your address data, use the Global Address Cleanse transform after the Global Suggestion Lists transform in the data flow.
Country ID	Identifies the country of destination for the record and outputs an ISO code. Use this transform before the Global Suggestion Lists transform in your data flow. (It is not necessary to place the Country ID transform before the Global Address Cleanse or the USA Regulatory Address Cleanse transforms.)

Parent topic: [How address cleanse works \[page 484\]](#)

Related Information

[Input and output data and field classes \[page 488\]](#)

18.6.5.1.2 Input and output data and field classes

Use address cleanse input and output fields based on the field format and the type of data the field contains.

Use multiline or discrete formats for your input and output fields, or use a combination of both. Also designate the type of output information for output fields by setting field classes.

Input and output field format types

Address line format	Description
Multiline	Map input address elements to multiline fields so that the software outputs address elements in the same order as they are input. • Software applies intelligent abbreviation when data exceeds the maximum field length. • Software capitalizes and standardizes output data based on options selected in transform.
Discrete	Map input address elements into smaller address elements for output. • Outputs additional fields for data generated by the software, such as error and status codes. • Software does not apply intelligent abbreviation for data that exceeds the maximum field length. • Software capitalizes and standardizes some output components based on options selected in transform.

For each output field in the USA Regulatory Address Cleanse and Global Address Cleanse transforms, select a generated address class type and a generated field class type:

- Field address class: Specifies the address class for the generated field.
- Field class: Specifies the type of information the transform includes in the output field.

Generated field address class and related field classes

Address class type	Description	Field class type
Delivery	Generates output fields that reflect the address that the software uses when it attempts to assign an address.	• Parsed: Outputs the parsed value with some standardization applied to the following fields: <i>Locality</i>, <i>Region</i>, and <i>Postcode</i>. • Best: Outputs the parsed data when the transform cannot assign the address. Outputs the corrected data when the transform can assign the address. • Corrected: Outputs the complete and correct value from the directories and standardizes the value based on set standardization options and other options. Output is blank when the transform cannot find the address in the directory and the address is unassigned.

Address class type	Description	Field class type
Dual	Generates output fields that reflect the address elements that the software does not use when it attempts to assign an address. Applicable for input records that may contain both a street and postal address.	<i>Parsed</i> , <i>Best</i> , and <i>Corrected</i> : Contains the DUAL address details that are available on input.
Official	Generates output fields in the form of the data preferred by the Postal Authority.	• <i>Parsed</i>: Outputs the parsed value with some standardization applied. • <i>Best</i>: Outputs the parsed data when the transform cannot assign the address. Outputs the corrected data when the transform can assign the address. • <i>Corrected</i>: Outputs the complete and correct value from the directories and standardizes the value based on set standardization options and other options. Output is blank when the transform cannot find the address in the directory and the address is unassigned.

Parent topic: [How address cleanse works \[page 484\]](#)

Related Information

[Address Cleanse transforms \[page 485\]](#)

18.6.5.2 Address cleanse reports

Both USA Regulatory Address Cleanse and Global Address Cleanse transforms enable you to generate certified postal reports.

Many of the reports generated by the software include reports that are required by the US Postal Service or other global postal authorities.

The USA Regulatory Address Cleanse transform provides data for the creation of the USPS Form 3553, which is required for CASS, and the NCOALink Summary Report. The Global Address Cleanse transform provides data for the creation of the Canada Post SERP report, the Australia Post AMAS report, and the New Zealand Post SOA Report.

Most postal and regulatory reports require that the software be certified by the respective postal authority on a periodic basis.

Parent topic: [Address Cleanse \[page 483\]](#)

Related Information

[How address cleanse works \[page 484\]](#)
[Preparing your input data \[page 490\]](#)
[Determining which transform\(s\) to use \[page 492\]](#)
[Identifying the country of destination \[page 497\]](#)
[Set up the reference files \[page 498\]](#)
[Defining the standardization options \[page 501\]](#)
[Process Japanese addresses \[page 503\]](#)
[Process Chinese addresses \[page 514\]](#)
[Supported countries \(Global Address Cleanse\) \[page 519\]](#)
[New Zealand certification \[page 522\]](#)
[Global Address Cleanse Suggestion List option \[page 527\]](#)
[Global Suggestion List transform \[page 528\]](#)
[Address assignment for buildings \[page 530\]](#)

18.6.5.3 Preparing your input data

Before you start with address cleansing, decide what kind of address line format to input.

Both the USA Regulatory Address Cleanse transform and the Global Address Cleanse transform accept input data in the same way.

Caution

The USA Regulatory Address Cleanse transform does not process Unicode data that has characters outside the Latin1 code page and character values greater than 255. Instead, the transform sends the input record to the corresponding standardized output field without any processing. Further, the transform does not populate any other output fields, such as component, for that record. If your Unicode database has valid U.S. addresses from the Latin1 character set, the USA Regulatory Address Cleanse transform processes as usual.

Accepted address line formats

The following table lists examples of multiline and multiline hybrid formats.

Note

When a multiline or hybrid format has more than one field, such as Multiline1-12, you do not have to use all of the fields. However, the transform requires that you start with the first field, Multiline1 for example, and continue using the fields consecutively. For example, you cannot skip numbers from Multiline1 to Multiline3.

❖ Example

Multiline and multiline hybrid formats

Example 1	Example 2	Example 3	Example 4	Example 5
Multiline1	Multiline1	Multiline1	Multiline1	Multiline1
Multiline2	Multiline2	Multiline2	Multiline2	Multiline2
Multiline3	Multiline3	Multiline3	Multiline3	Multiline3
Multiline4	Multiline4	Multiline4	Multiline4	Multiline4
Multiline5	Multiline5	Locality3	Multiline5	Multiline5
Multiline6	Multiline6	Locality2	Locality2	Multiline6
Multiline7	Multiline7	Locality1	Locality1	Locality1
Multiline8	Lastline	Region1	Region1	Region1
Country (Optional)	Country (Optional)	Postcode (Global) or Postcode1 (USA Reg.)	Postcode (Global) or Postcode1 (USA Reg.)	Postcode (Global) or Postcode1 (USA Reg.)
		Country (Optional)	Country (Optional)	Country (Optional)

The following table shows examples of accepted discrete line formats.

❖ Example

Discrete line formats

Example 1	Example 2	Example 3	Example 4
Address_Line	Address_Line	Address_Line	Address_Line
Lastline	Locality3 (Global)	Locality2	Locality1
Country (Optional)	Locality2	Locality1	Region1
	Locality1	Region1	Postcode (Global) or Post- code1 (USA Reg.)
	Region1	Postcode (Global) or Post- code1 (USA Reg.)	Country (Optional)
	Postcode (Global) or Post- code1 (USA Reg.)	Country (Optional)	
	Country (Optional)		

Parent topic: [Address Cleanse \[page 483\]](#)

Related Information

[How address cleanse works \[page 484\]](#)

[Address cleanse reports \[page 489\]](#)
[Determining which transform\(s\) to use \[page 492\]](#)
[Identifying the country of destination \[page 497\]](#)
[Set up the reference files \[page 498\]](#)
[Defining the standardization options \[page 501\]](#)
[Process Japanese addresses \[page 503\]](#)
[Process Chinese addresses \[page 514\]](#)
[Supported countries \(Global Address Cleanse\) \[page 519\]](#)
[New Zealand certification \[page 522\]](#)
[Global Address Cleanse Suggestion List option \[page 527\]](#)
[Global Suggestion List transform \[page 528\]](#)
[Address assignment for buildings \[page 530\]](#)

18.6.5.4 Determining which transform(s) to use

You can choose from a variety of address cleanse transforms based on what you want to do with your data. There are transforms for cleansing global and/or U.S. address data, cleansing based on USPS regulations, using business rules to cleanse data and cleansing global address data transactionally.

[Cleansing global address data \[page 493\]](#)

To cleanse your address data for any of the software-supported countries, use the Global Address Cleanse transform.

[Cleansing U.S. data only \[page 494\]](#)

[Cleansing U.S. data and global data \[page 495\]](#)

[Cleansing address data using multiple business rules \[page 495\]](#)

[Cleansing your address data transactionally \[page 496\]](#)

Parent topic: [Address Cleanse \[page 483\]](#)

Related Information

[How address cleanse works \[page 484\]](#)
[Address cleanse reports \[page 489\]](#)
[Preparing your input data \[page 490\]](#)
[Identifying the country of destination \[page 497\]](#)
[Set up the reference files \[page 498\]](#)
[Defining the standardization options \[page 501\]](#)
[Process Japanese addresses \[page 503\]](#)
[Process Chinese addresses \[page 514\]](#)
[Supported countries \(Global Address Cleanse\) \[page 519\]](#)
[New Zealand certification \[page 522\]](#)

[Global Address Cleanse Suggestion List option \[page 527\]](#)
[Global Suggestion List transform \[page 528\]](#)
[Address assignment for buildings \[page 530\]](#)
[Cleansing global address data \[page 493\]](#)
[Cleansing U.S. data only \[page 494\]](#)
[Cleansing U.S. data and global data \[page 495\]](#)
[Cleansing address data using multiple business rules \[page 495\]](#)
[Cleansing your address data transactionally \[page 496\]](#)

18.6.5.4.1 Cleansing global address data

To cleanse your address data for any of the software-supported countries, use the Global Address Cleanse transform.

Select the engines to use in the transform based on the goals for your processing:

- To generate the Canada Post certified SERP (Software Evaluation and Recognition Program), enable the Canada engine.
- To generate the Australia Post AMAS (Address Matching Approval System), enable the Global Address.
- To process a mix of countries, including USA addresses, enable the USA engine in combination with the other applicable engines.

→ Tip

Cleansing U.S. data with the USA Regulatory Address Cleanse transform is usually faster than with the Global Address Cleanse transform and USA engine. This scenario is usually true even if you end up needing both transforms.

Also use the Global Address Cleanse transform with the Canada, USA, and—or Global Address engines in a real-time data flow to create suggestion lists for those countries.

The Global Address Cleanse transform supports many countries. However, there are some limitations:

- The level of address correction varies by country and by the engine that you use. Complete coverage of all addresses in a country is not guaranteed.
- For the Global Address engine, country support depends on the sets of postal directories you purchase.
- For Japan, the transform bases the assignment level on data provided by the Ministry of Public Management Home Affairs, Posts, and Telecommunications (MPT).
- During Country ID processing, the transform identifies many countries. However, the Global Address Cleanse transform engines may not provide address correction for all of those countries.

Start with a sample transform configuration

The software includes a variety of Global Address Cleanse sample transform configurations (which include at least one engine) that you can copy to use for a project.

Parent topic: [Determining which transform\(s\) to use \[page 492\]](#)

Related Information

[Cleansing U.S. data only \[page 494\]](#)

[Cleansing U.S. data and global data \[page 495\]](#)

[Cleansing address data using multiple business rules \[page 495\]](#)

[Cleansing your address data transactionally \[page 496\]](#)

[Supported countries \(Global Address Cleanse\) \[page 519\]](#)

[Cleansing U.S. data and global data \[page 495\]](#)

[Reference Guide: Transforms, Data Quality transforms, Transform configurations](#)

18.6.5.4.2 Cleansing U.S. data only

To cleanse U.S. address data, use the USA Regulatory Address Cleanse transform for the best results. With this transform, and with DPV, LACSLink, and SuiteLink enabled, you can produce a CASS-certified mailing and produce a USPS Form 3553. If you do not intend to process CASS-certified lists, you should still use the USA Regulatory Address Cleanse transform for processing your U.S. data. Using the USA Regulatory Address Cleanse transform on U.S. data is more efficient than using the Global Address Cleanse transform.

With the USA Regulatory Address Cleanse transform you can add additional information to your data such as DSF2, EWS, eLOT, NCOALink, and RDI. And you can process records one at a time by using suggestion lists.

Start with a sample transform configuration

The software includes a variety of USA Regulatory Address Cleanse sample transform configurations that can help you set up your projects.

Parent topic: [Determining which transform\(s\) to use \[page 492\]](#)

Related Information

[Cleansing global address data \[page 493\]](#)

[Cleansing U.S. data and global data \[page 495\]](#)

[Cleansing address data using multiple business rules \[page 495\]](#)

[Cleansing your address data transactionally \[page 496\]](#)

[Reference Guide: Transforms, Data Quality transforms, Transform configurations](#)

[How suggestion lists work \[page 611\]](#)

18.6.5.4.3 Cleansing U.S. data and global data

What should you do when you have U.S. addresses that need to be certified and also addresses from other countries in your database? In this situation, you should use both the Global Address Cleanse transform and the USA Regulatory Address Cleanse transform in your data flow.

→ Tip

Even if you are not processing U.S. data for USPS certification, you may find that cleansing U.S. data with the USA Regulatory Address Cleanse transform is faster than with the Global Address Cleanse transform and USA engine.

Parent topic: [Determining which transform\(s\) to use \[page 492\]](#)

Related Information

[Cleansing global address data \[page 493\]](#)

[Cleansing U.S. data only \[page 494\]](#)

[Cleansing address data using multiple business rules \[page 495\]](#)

[Cleansing your address data transactionally \[page 496\]](#)

18.6.5.4.4 Cleansing address data using multiple business rules

When you have two addresses intended for different purposes (for example, a billing address and a shipping address), you should use two of the same address cleanse transforms in a data flow.

One or two engines?

When you use two Global Address Cleanse transforms for data from the same country, they can share an engine. You do not need to have two engines of the same kind. If you use one engine or two, it does not affect the overall processing time of the data flow.

In this situation, however, you may need to use two separate engines (even if the data is from the same country). Depending on your business rules, you may have to define the settings in the engine differently for a billing address or for a shipping address. For example, in the Standardization Options group, the Output Country Language option can convert the data used in each record to the official country language or it can preserve the language used in each record. For example, you may want to convert the data for the shipping address but preserve the data for the billing address.

Parent topic: [Determining which transform\(s\) to use \[page 492\]](#)

Related Information

[Cleansing global address data \[page 493\]](#)

[Cleansing U.S. data only \[page 494\]](#)

[Cleansing U.S. data and global data \[page 495\]](#)

[Cleansing your address data transactionally \[page 496\]](#)

18.6.5.4.5 Cleansing your address data transactionally

The Global Suggestion Lists transform, best used in transactional projects, is a way to complete and populate addresses with minimal data, or it can offer suggestions for possible matches. For example, the Marshall Islands and the Federated States of Micronesia were recently removed from the USA Address directory. Therefore, if you previously used the USA engine, you'll now have to use the Global Address engine. The Global Suggestion Lists transform can help identify that these countries are no longer in the USA Address directory.

This easy address-entry system is ideal in call center environments or any transactional environment where cleansing is necessary at the point of entry. It's also a beneficial research tool when you need to manage bad addresses from a previous batch process.

Place the Global Suggestion Lists transform after the Country ID transform and before a Global Address Cleanse transform that uses a Global Address, Canada, and/or USA engine.

Integrating functionality

Global Suggestion Lists functionality is designed to be integrated into your own custom applications via the Web Service. If you are a programmer looking for details about how to integrate this functionality, see "Integrate Global Suggestion Lists" in the *Integrator Guide*.

Start with a sample transform configuration

Data Quality includes a Global Suggestion Lists sample transform that can help you when setting up a project.

Parent topic: [Determining which transform\(s\) to use \[page 492\]](#)

Related Information

[Cleansing global address data \[page 493\]](#)

[Cleansing U.S. data only \[page 494\]](#)

[Cleansing U.S. data and global data \[page 495\]](#)

[Cleansing address data using multiple business rules \[page 495\]](#)

[How suggestion lists work \[page 611\]](#)

18.6.5.5 Identifying the country of destination

The Global Address Cleanse transform includes Country ID processing. Therefore, you do not need to place a Country ID transform before the Global Address Cleanse transform in your data flow.

In the Country ID Options option group of the Global Address Cleanse transform, you can define the country of destination or define whether you want to run Country ID processing.

Constant country

If all of your data is from one country, such as Australia, you do not need to run Country ID processing or input a discrete country field. You can tell the Global Address Cleanse transform the country and it will assume all records are from this country (which may save processing time).

Assign default

You'll want to run Country ID processing if you are using two or more of the engines and your input addresses contain country data (such as the two-character ISO code or a country name), or if you are using only one engine and your input source contains many addresses that cannot be processed by that engine. Addresses that cannot be processed are not sent to the engine. The transform will use the country you specify in this option group as a default.

Parent topic: [Address Cleanse \[page 483\]](#)

Related Information

[How address cleanse works \[page 484\]](#)

[Address cleanse reports \[page 489\]](#)

[Preparing your input data \[page 490\]](#)

[Determining which transform\(s\) to use \[page 492\]](#)

[Set up the reference files \[page 498\]](#)

[Defining the standardization options \[page 501\]](#)

[Process Japanese addresses \[page 503\]](#)

[Process Chinese addresses \[page 514\]](#)

[Supported countries \(Global Address Cleanse\) \[page 519\]](#)

[New Zealand certification \[page 522\]](#)

[Global Address Cleanse Suggestion List option \[page 527\]](#)

[Global Suggestion List transform \[page 528\]](#)

[Address assignment for buildings \[page 530\]](#)

[Setting a constant country \[page 521\]](#)

[Setting a default country \[page 521\]](#)

18.6.5.6 Set up the reference files

The USA Regulatory Address Cleanse transform and the Global Address Cleanse transform and engines rely on several directories (reference files) to cleanse your data.

When you use directories to standardize addresses and assign postal codes, you must use the most up-to-date directories, which requires you to periodically update your files with new directories.

SAP keeps you informed about current directory dates, and alerts you when you need to download newer directories.

To correct addresses and assign postal codes, the address cleanse transforms rely on databases called postal directories.

The Global Address Cleanse transform relies on global directories based on the chosen domain.

Directories help extend address cleansing beyond the basic parsing and standardizing.

[Directories \[page 499\]](#)

To correct addresses and assign codes, the address cleanse transforms relies on address directories.

[Directory file locations \[page 500\]](#)

Set directory file locations in the transform's *Reference Files* option group.

[Directory files substitution values \[page 500\]](#)

Use substitution values to set a constant value for your directory locations.

[Viewing directory expiration dates in the trace log \[page 501\]](#)

Steps to set up the trace log with directory expiration dates.

Parent topic: [Address Cleanse \[page 483\]](#)

Related Information

[How address cleanse works \[page 484\]](#)

[Address cleanse reports \[page 489\]](#)

[Preparing your input data \[page 490\]](#)

[Determining which transform\(s\) to use \[page 492\]](#)

[Identifying the country of destination \[page 497\]](#)

[Defining the standardization options \[page 501\]](#)

[Process Japanese addresses \[page 503\]](#)

[Process Chinese addresses \[page 514\]](#)

[Supported countries \(Global Address Cleanse\) \[page 519\]](#)
[New Zealand certification \[page 522\]](#)
[Global Address Cleanse Suggestion List option \[page 527\]](#)
[Global Suggestion List transform \[page 528\]](#)
[Address assignment for buildings \[page 530\]](#)
[Supported countries \(Global Address Cleanse\) \[page 519\]](#)
[Process Japanese addresses \[page 503\]](#)
[Process Chinese addresses \[page 514\]](#)
[New Zealand certification \[page 522\]](#)
[USPS certifications](#)

18.6.5.6.1 Directories

To correct addresses and assign codes, the address cleanse transforms relies on address directories.

These directories contain specific types of information related to what processes you want performed on your addresses. For example, the United States Postal Service has directories with address information about every address in the United States and its territories. And the software uses many other directories issued by other postal authorities such as Canada, New Zealand, and Japan.

i Note

Address cleanse transforms support many countries, however, when you use the Global Address Cleanse transform, there may be some countries that are not supported by official directories.

Besides the basic address directories, there are many specialized directories. For example, the USA Regulatory Address Cleanse transform uses the following specialized directories:

- DPV®
- DSF2®
- Early Warning System (EWS)
- eLOT®
- GeoCensus
- LACSLink®
- NCOALink®
- RDI™
- SuiteLink™
- Z4Change

These specialized directories help extend US address cleansing beyond the basic parsing and standardizing.

Parent topic: [Set up the reference files \[page 498\]](#)

Related Information

[Directory file locations \[page 500\]](#)

[Directory files substitution values \[page 500\]](#)

[Viewing directory expiration dates in the trace log \[page 501\]](#)

18.6.5.6.2 Directory file locations

Set directory file locations in the transform's *Reference Files* option group.

Your system administrator should have already installed the necessary files to the appropriate locations based on your company's needs.

Caution

Incompatible or out-of-date directories can render the software unusable. The system administrator must install weekly, monthly or bimonthly directory updates for the USA Regulatory Address Cleanse Transform; monthly directory updates for the Australia and Canada engines; and quarterly directory updates for the Global Address engine to ensure that they are compatible with the current software.

Parent topic: [Set up the reference files \[page 498\]](#)

Related Information

[Directories \[page 499\]](#)

[Directory files substitution values \[page 500\]](#)

[Viewing directory expiration dates in the trace log \[page 501\]](#)

18.6.5.6.3 Directory files substitution values

Use substitution values to set a constant value for your directory locations.

If you start with a sample transform, the *Reference Files* options are already completed with a substitution variable by default. For example, the variable \$\$RefFilesAddressCleanse points to the reference data folder of the software directory by default.

Note

You can change that location by editing the substitution file associated with the data flow. This change is made for every data flow that uses that substitution file.

Parent topic: [Set up the reference files \[page 498\]](#)

Related Information

[Directories \[page 499\]](#)

[Directory file locations \[page 500\]](#)

[Viewing directory expiration dates in the trace log \[page 501\]](#)

[Overview of substitution parameters \[page 720\]](#)

18.6.5.6.4 Viewing directory expiration dates in the trace log

Steps to set up the trace log with directory expiration dates.

To include directory expiration information in the trace log, perform the following steps.

1. Right click on the applicable job icon in Designer and select *Execute*.
2. In the *Execution Properties* window, open the *Execution Options* tab (it should already be open by default).
3. Select *Print all trace messages*.

Parent topic: [Set up the reference files \[page 498\]](#)

Related Information

[Directories \[page 499\]](#)

[Directory file locations \[page 500\]](#)

[Directory files substitution values \[page 500\]](#)

[Access logs \[page 761\]](#)

18.6.5.7 Defining the standardization options

Standardization changes the way the data is presented after an assignment has been made. The type of change depends on the options that you define in the transform. These options include casing, punctuation, sequence, abbreviations, and much more. It helps ensure the integrity of your databases, makes mail more deliverable, and gives your communications with customers a more professional appearance.

For example, the following address was standardized for capitalization, punctuation, and postal phrase (route to RR).

Input	Output
Multiline1 = route 1 box 44a	Address_Line = RR 1 BOX 44A
Multiline2 = stodard wisc	Locality1 = STODDARD
	Region1 = WI
	Postcode1 = 54658

Global Address Cleanse transform

In the Global Address Cleanse transform, you set the standardization options in the Standardization Options option group.

You can standardize addresses for all countries and/or for individual countries (depending on your data). For example, you can have one set of French standardization options that standardize addresses within France only, and another set of Global standardization options that standardize all other addresses.

USA Regulatory Address Cleanse transform

If you use the USA Regulatory Address Cleanse transform, you set the standardization options on the [Options](#) tab in the Standardization Options section.

Parent topic: [Address Cleanse \[page 483\]](#)

Related Information

[How address cleanse works \[page 484\]](#)

[Address cleanse reports \[page 489\]](#)

[Preparing your input data \[page 490\]](#)

[Determining which transform\(s\) to use \[page 492\]](#)

[Identifying the country of destination \[page 497\]](#)

[Set up the reference files \[page 498\]](#)

[Process Japanese addresses \[page 503\]](#)

[Process Chinese addresses \[page 514\]](#)

[Supported countries \(Global Address Cleanse\) \[page 519\]](#)

[New Zealand certification \[page 522\]](#)

[Global Address Cleanse Suggestion List option \[page 527\]](#)

[Global Suggestion List transform \[page 528\]](#)

[Address assignment for buildings \[page 530\]](#)

[Reference Guide: Transforms, Global Address Cleanse transform options \(Standardization options\)](#)

18.6.5.8 Process Japanese addresses

The Global Address Cleanse transform and the Global Address engine parses and normalizes Japanese addresses for address cleansing and data matching.

i Note

The Japan engine supports only kanji and katakana data. The engine does not support Latin data.

A significant portion of the address parsing capability relies on the Japanese address database. The software has data from the Ministry of Public Management, Home Affairs, Posts, and Telecommunications (MPT) and additional data sources. The enhanced address database consists of a regularly updated government database that includes regional postal codes mapped to localities.

[Standard Japanese address format \[page 504\]](#)

A typical Japanese address includes many components that are similar to USA addresses, and some components that are unique to Japanese addresses.

[Special Hokkaido regional formats \[page 510\]](#)

The Hokkaido region has two special address formats.

[Other special Japanese address formats \[page 511\]](#)

Japanese addresses have other special address formats such as accepted spellings and numbering.

[Sample Japanese address \[page 512\]](#)

The Global Address Cleanse transform processes the various parts of a Japanese address by parsing the address elements into address line fields, lastline fields, firm fields, and nonparsed fields.

Parent topic: [Address Cleanse \[page 483\]](#)

Related Information

[How address cleanse works \[page 484\]](#)

[Address cleanse reports \[page 489\]](#)

[Preparing your input data \[page 490\]](#)

[Determining which transform\(s\) to use \[page 492\]](#)

[Identifying the country of destination \[page 497\]](#)

[Set up the reference files \[page 498\]](#)

[Defining the standardization options \[page 501\]](#)

[Process Chinese addresses \[page 514\]](#)

[Supported countries \(Global Address Cleanse\) \[page 519\]](#)

[New Zealand certification \[page 522\]](#)

[Global Address Cleanse Suggestion List option \[page 527\]](#)

[Global Suggestion List transform \[page 528\]](#)

[Address assignment for buildings \[page 530\]](#)
[Standard Japanese address format \[page 504\]](#)
[Special Hokkaido regional formats \[page 510\]](#)
[Sample Japanese address \[page 512\]](#)

18.6.5.8.1 Standard Japanese address format

A typical Japanese address includes many components that are similar to USA addresses, and some components that are unique to Japanese addresses.

The following table contains typical Japanese address components.

Address component	Japanese	English	Output field(s)
Postal code	〒 654-0153	654-0153	<i>Postcode_Full</i>
Prefecture	兵庫県	Hyogo-ken	<i>Region1_Full</i>
City	神戸市	Kobe-shi	<i>Locality1_Full</i>
Ward	須磨区	Suma-ku	<i>Locality2_Full</i>
District	南落合	Minami Ochiai	<i>Locality3_Full</i>
Block number	1 丁目	1 chome	<i>Primary_Name_Full1</i>
Sub-block number	25 番地	25 banchi	<i>Primary_Name_Full2</i>
House number	2 号	2 go	<i>Primary_Number_Full</i>

An address may also include building name, floor number, and room number.

Parent topic: [Process Japanese addresses \[page 503\]](#)

Related Information

[Special Hokkaido regional formats \[page 510\]](#)
[Other special Japanese address formats \[page 511\]](#)
[Sample Japanese address \[page 512\]](#)
[Japanese address components \[page 505\]](#)

18.6.5.8.1.1 Japanese address components

Japanese addresses have many address components, some of which are subcomponents of bigger components.

❖ Example

A Japanese city is divided into wards. A ward is divided into districts.

The following table describes the Japanese address components.

Japanese address component	Description
Postal code	Japanese postal codes represent an area and a location type. Japanese postal codes use the following format: nnn-nnnn. • The first three digits represent the area. • The last four digits represent a location in the area: ◦ District ◦ Subdistrict ◦ Block ◦ Subblock ◦ Building ◦ Floor ◦ Company Postal authorities require postal codes written with Arabic numbers. The post office symbol 〒 is optional. Before 1998, the postal code consisted of 3 or 5 digits. Some older databases may still reflect the old system.
Prefecture	Prefectures are regions. Japan has forty-seven prefectures. You may omit the prefecture for some well known cities.
City	Japanese city names have the suffix 市 (-shi). In some parts of the Tokyo and Osaka regions, people omit the city name. In some island villages, they use the island name with a suffix 島 (-shima) in place of the city name. In some rural areas, they use the county name with suffix 郡 (-gun) in place of the city name.
Ward	A city is divided into wards. The ward name has the suffix 区 (-ku). The ward component is omitted for small cities, island villages, and rural areas that don't have wards.

Japanese address component	Description
District	A ward is divided into districts. When there is no ward, the small city, island village, or rural area is divided into districts. The district name may have the suffix 町 (-cho/-machi), but it is sometimes omitted. 町 has two possible pronunciations, but only one is correct for a particular district. In very small villages, people use the village name with suffix 村 (-mura) in place of the district. When a village or district is on an island with the same name, the island name is often omitted.
Sub-district	Primarily in rural areas, a district may be divided into sub-districts, marked by the prefix 字 (aza-). A sub-district may be further divided into sub-districts that are marked by the prefix 小字 (koaza-), meaning small aza. koaza may be abbreviated to aza. A sub-district may also be marked by the prefix 大字 (oaza-), which means large aza. Oaza may also be abbreviated to aza. Here are the possible combinations: • oaza • aza • oaza and aza • aza and koaza • oaza and koaza i Note Addresses frequently omit the characters 大字 (oaza-), 字(aza-), and 小字 (koaza-).
Sub-district parcel	A sub-district aza may be divided into numbered sub-district parcels, which are marked by the suffix 部 (-bu), meaning <i>piece</i>. The character 部 is frequently omitted. Parcels can be numbered in several ways: • Arabic numbers (1, 2, 3, 4, and so on) 石川県七尾市松百町 8 部 3 番地 1 号 • Katakana letters in iroha order (イ, 口, ハ, ニ, and so on) 石川県小松市里川町ナ部 23 番地 • Kanji numbers, which is very rare (甲, 乙, 丙, 丁, and so on) 愛媛県北条市上難波甲部 311 番地

Japanese address component	Description
Sub-division	A rural district or sub-district (oaza/aza/koaza) is sometimes divided into sub-divisions, marked by the suffix 地割 (-chiwari) which means division of land. The optional prefix is 第 (dai-). The following address examples show sub-divisions: 岩手県久慈市旭町 10 地割 1 番地 岩手県久慈市旭町第 10 地割 1 番地
Block number	A district is divided into blocks. The block number includes the suffix 丁目 (-chome). Districts usually have between 1 and 5 blocks, but they can have more. The block number may be written with a Kanji number. Japanese addresses do not include a street name. 東京都渋谷区道玄坂 2 丁目 2 5 番地 1 2 号 東京都渋谷区道玄坂二丁目 2 5 番地 1 2 号
Sub-block number	A block is divided into sub-blocks. The sub-block name includes the suffix 番地 (-banchi), which means numbered land. The suffix 番地 (-banchi) may be abbreviated to just 番 (-ban).
House number	Each house has a unique house number. The house number includes the suffix 号 (-go), which means number.

Block, subblock, and house number variations

Block, subblock, and house number data may vary. The following table contains possible variations.

Variant	Description
Dashes	The suffix markers 丁目(chome), 番地(banchi), and 号(go) may be replaced with dashes. 東京都文京区湯島 2 丁目 18 番地 12 号 東京都文京区湯島 2-18-12 Sometimes block, sub-block, and house number are combined or omitted. 東京都文京区湯島 2 丁目 18 番 12 号 東京都文京区湯島 2 丁目 18 番地 12 東京都文京区湯島 2 丁目 18-12

Variant	Description
No block number	Sometimes the block number is omitted. For example, the following address is a ward of Tokyo. It has numbered districts and no block numbers. 東京都 千代田区 二番町 9 番地 6 号 二番町 means district number 2.
Building names	Building names can vary in the following ways: - Names of apartments or buildings are often included after the house number. - When a building name includes the name of the district, the district name is often omitted. - When a building is well known, the block, subblock, and house number are often omitted. - When a building name is long, it may be abbreviated or written using its acronym with English letters. For a list with descriptions, see Common building name suffixes [page 508].
Building numbers	- Numbers such as room numbers and apartment numbers follow the building name. - Building numbers may include the suffix 号室 (-goshitsu). - Floor numbers above ground level may include the suffix 階 (-kai) or the letter F. - Floor numbers below ground level may include the suffix 地下<n>階 (chika <n> kai) or the letters B<n>F (where <n> represents the floor number). - An apartment complex may include multiple buildings called Building A, Building B, and so on, marked by the suffix 棟 (-tou). For address examples include building numbers see Japanese building number examples [page 509].

Related Information

[Standard Japanese address format \[page 504\]](#)

18.6.5.8.1.2 Common building name suffixes

Japanese building names can have one of many suffixes.

For Japanese addresses, names of apartments or buildings are often included after the house number. When a building name includes the name of the district, the district name is often omitted. When a building is well known, the block, sub-block, and house number are often omitted. When a building name is long, it may be abbreviated or written using its acronym with English letters.

The following table contains a list of the common suffixes.

Suffix	Romanized	Translation
ビルディング	birudingu	building
ビルヂング	birudingu	building
ビル	biru	building
センター	senta-	center
プラザ	puraza	plaza
パーク	pa-ku	park
タワー	tawa-	tower
会館	kaikan	hall
棟	tou	building (unit)
庁舎	chousha	government office building
マンション	manshon	condominium
団地	danchi	apartment complex
アパート	apa-to	apartment
荘	sou	villa
住宅	juutaku	housing
社宅	shataku	company housing
官舎	kansha	official residence

18.6.5.8.1.3 Japanese building number examples

Table that contains sample Japanese addresses that include building numbers.

Building area	Example
Third floor above ground	東京都千代田区二番町 9 番地 6 号 バウエプタ 3 F
Second floor below ground	東京都渋谷区道玄坂 2-25-12 シティバンク地下 2 階
Building A Room 301	兵庫県神戸市須磨区南落合 1-25-10 須磨パークヒルズ A 棟 301 号室
Building A Room 301	兵庫県神戸市須磨区南落合 1-25-10 須磨パークヒルズ A-301

Related Information

[Japanese address components \[page 505\]](#)

18.6.5.8.2 Special Hokkaido regional formats

The Hokkaido region has two special address formats.

Hokkaido regional format

Format	Description
Super block	A special super-block format exists only in the Hokkaido prefecture. A super-block, marked by the suffix 条 (-joh), is one level larger than the block. The super-block number or the block number may contain a directional 北 (north), 南 (south), 東 (east), or 西 (west). The following address example shows a super-block 4 Joh. 北海道札幌市西区二十四軒 4 条 4 丁目 1 3 番地 7 号
Numbered subdistricts	A subdistrict name may be marked with the suffix 線 (-sen) meaning number instead of the suffix 字 (-aza). When a subdistrict has a 線 suffix, the block may have the suffix 号 (-go), and the house number has no suffix. The following address contains first the sub-district 4 sen and then a numbered block 5 go: 北海道旭川市西神楽 4 線 5 号 3 番地 1 1

Parent topic: [Process Japanese addresses \[page 503\]](#)

Related Information

[Standard Japanese address format \[page 504\]](#)

[Other special Japanese address formats \[page 511\]](#)

[Sample Japanese address \[page 512\]](#)

18.6.5.8.3 Other special Japanese address formats

Japanese addresses have other special address formats such as accepted spellings and numbering.

Format	Description
Accepted spelling	Names of cities, districts, and so on, can have multiple accepted spellings because there are multiple accepted ways to write certain sounds in Japanese.
Accepted numbering	When the block, subblock, house number, or district contains a number, the number may be written in Arabic or Kanji. For example, 二番町 means district number 2, and in the following example it is for Niban-cho. 東京都千代田区二番町九番地六号
P.O. Box addresses	P.O. Box addresses contain the postal code, Locality1, prefecture, the name of the post office, the box marker, and the box number. i Note The Global Address Cleanse transform recognizes P.O. box addresses that are located in the Large Organization Postal Code (LOPC) database only. The address may be in one of the following formats: • Prefecture, Locality1, post office name, box marker (私書箱), and P.O. box number. • Postal code, prefecture, Locality1, post office name, box marker (私書箱), and P.O. box number. The following address example shows a P.O. Box address: The Osaka Post Office Box marker #1 大阪府大阪市大阪支店私書箱 1 号
Large Organization Postal Code (LOPC) format	The Postal Service may assign a unique postal code to a large organization, such as the customer service department of a major corporation. An organization may have up to two unique postal codes depending on the volume of mail it receives. The address may be in one of the following formats: • Address, company name • Postal code, address, company name The following is an example of an address in a LOPC address format. 100-8798 東京都千代田区霞が関 1 丁目 3 - 2 日本郵政 株式会社

Parent topic: [Process Japanese addresses \[page 503\]](#)

Related Information

- Standard Japanese address format [page 504]
- Special Hokkaido regional formats [page 510]
- Sample Japanese address [page 512]

18.6.5.8.4 Sample Japanese address

The Global Address Cleanse transform processes the various parts of a Japanese address by parsing the address elements into address line fields, lastline fields, firm fields, and nonparsed fields.

❖ Example	
Input address	
0018521 北海道札幌市北区北十条西 1 丁目 12 番地 3 号創生ビル 1 階 101 号室札幌私書箱センター	
Address-line fields	
Primary_Name1	1
Primary_Type1	丁目
Primary_Name2	12
Primary_Type2	番地
Primary_Number	3
Primary_Number_Description	号
Building_Name1	創生ビル
Floor_Number	1
Floor_Description	階
Unit_Number	101
Unit_Description	号室
Primary_Address	1 丁目 12 番地 3 号
Secondary_Address	創生ビル 1 階 101 号室
Primary_Secondary_Address	1 丁目 12 番地 3 号 創生ビル 1 階 101 号室

Last line fields

<i>Country</i>	日本
<i>ISO_Country_Code_3Digit</i>	392
<i>ISO_Country_Code_2Char</i>	JP
<i>Postcode1</i>	001
<i>Postcode2</i>	8521
<i>Postcode_Full</i>	001-8521
<i>Region1</i>	北海
<i>Region1_Description</i>	道
<i>Locality1_Name</i>	札幌
<i>Locality1_Description</i>	市
<i>Locality2_Name</i>	北
<i>Locality2_Description</i>	区
<i>Locality3_Name</i>	北十条西
<i>Lastline</i>	001-8521 北海道 札幌市 北区 北十条西

Firm

<i>Firm</i>	札幌私書箱センター

Non-parsed fields

<i>Status_Code</i>	S0000
<i>Assignment_Type</i>	F
<i>Address_Type</i>	S

Parent topic: [Process Japanese addresses \[page 503\]](#)

Related Information

[Standard Japanese address format \[page 504\]](#)

[Special Hokkaido regional formats \[page 510\]](#)

[Other special Japanese address formats \[page 511\]](#)

18.6.5.9 Process Chinese addresses

The Global Address engine in the Global Address Cleanse transform parses Chinese addresses.

The primary purpose of Global Address Cleanse transform and engine is to parse and normalize Chinese addresses for data matching and cleansing applications.

[Chinese address format \[page 514\]](#)

Chinese Addresses start with the postal code, followed by the largest administrative region, such as province, and continues down to the smallest unit, such as room number and mail receiver.

[Sample Chinese address \[page 518\]](#)

The Global Address Cleanse transform processes the various parts of a Chinese address by parsing the address elements into address line fields, lastline fields, and nonparsed fields.

Parent topic: [Address Cleanse \[page 483\]](#)

Related Information

[How address cleanse works \[page 484\]](#)

[Address cleanse reports \[page 489\]](#)

[Preparing your input data \[page 490\]](#)

[Determining which transform\(s\) to use \[page 492\]](#)

[Identifying the country of destination \[page 497\]](#)

[Set up the reference files \[page 498\]](#)

[Defining the standardization options \[page 501\]](#)

[Process Japanese addresses \[page 503\]](#)

[Supported countries \(Global Address Cleanse\) \[page 519\]](#)

[New Zealand certification \[page 522\]](#)

[Global Address Cleanse Suggestion List option \[page 527\]](#)

[Global Suggestion List transform \[page 528\]](#)

[Address assignment for buildings \[page 530\]](#)

18.6.5.9.1 Chinese address format

Chinese Addresses start with the postal code, followed by the largest administrative region, such as province, and continues down to the smallest unit, such as room number and mail receiver.

When people send mail between prefectures, they often include the largest administrative region in the address. Buildings along the street are numbered sequentially, sometimes with odd numbers on one side and

even numbers on the other side. In some instances, both odd and even numbers are on the same side of the street.

Chinese addresses have many components that identify various parts of an address. The following table contains descriptions of Chinese address components.

Format	Description
Postal Code	A 6-digit number to identify the target delivery point of the address. Postal codes often have the prefix 邮编.
Country	中华人民共和国 is People's Republic of China, which is the full name for China. China often uses the words “中国 (PRC)” as an abbreviation of the country name. For mail delivered within China, the domestic addresses often omit the country name of the target address.
Province	Similar to a state in the United States. China has 34 province-level divisions, including the following divisions: - Provinces (省 shěng) - Autonomous regions (自治区 zìzhìqū) - Municipalities (直辖市 zhíxiáshì) - Special administrative regions (特别行政区 tèbié xíngzhèngqū)
Prefecture	Prefectures are similar to what a state is in the US. Prefecture-level divisions are the second level of the administrative structure. China has 34 prefectures. Each prefecture name is followed by a suffix, which is one of the following four words: - 地区 dìqū: Prefectures - 自治州 zìzhìzhōu: Autonomous prefectures - 地级市 dìjīshì: Prefecture-level cities - 盟 méng: Leagues Some municipalities include the race name.
County	A sub-division of Prefecture that includes: - 县 xiàn: Counties - 自治县 zìzhìxiàn: Autonomous counties - 县级市 xiànjíshì: County-level cities - 市辖区 shìxiáqū: Districts - 旗 qí: Banners - 自治旗 zìzhìqí: Autonomous banners - 林区 línqū: Forestry areas - 特区 tèqū: Special districts
Township	Township level division that includes: - 乡 xiāng: Townships - 民族乡 mínzúxiāng: Ethnic townships - 镇 zhèn: Towns - 街道办事处 jiēdào bànshì chù: Subdistricts - 区公所 qūgōngsuǒ: District public offices - 苏木 sūmù: Sumu - 民族苏木 mínzúsūmù: Ethnic sumu

Format	Description
Village	Villages include the following components: - 社区居民委员会 jūmínwěiyuánhùi: Neighborhood committees - 社区: Neighborhoods or communities - 村民委员会 cūnmínwěiyuánhùi: Village committees - 村民小组 cūnmínxiǎozǔ: Village groups - 行政村 xíngzhèngcūn: Administrative villages
Street information	Specifies the delivery point where the mail receiver can be found. In China, the street information often has the form of [Town name] -> [Village name] -> Street (Road) name -> [Block number] -> House number. The components within the brackets [] are optional components in the address, and may often be omitted. The following list describes the street information components: - Town name: The town name is usually followed by the suffix 镇, 乡, or 堡. - Village name: The village name is usually followed by the suffix 村, 新村. - Street (Road) name: The street (road) name is usually followed by one of these suffixes 路, 大道, 街, 大街, 里. - Block number: The block number is usually followed by one of these suffixes 弄, 巷, 厅, 胡同. - House number: The house number is followed by the suffix 号. The house name is a unique number within the block or street/road.
Residential community	May be used for mail delivery, especially for some famous residential communities in major cities. The street name and house number might be omitted. The residential community does not have a naming standard. The residential community is not strictly required to be followed by a typical marker. However, the residential community is often followed by the typical suffixes, such as 新村, 小区, and so on.
Building name	Building name is often followed by a building marker such as 大厦 or 大楼, even though it is not strictly required. For example, 中华大厦. Building name in the residential communities is often represented by a number with a suffix of 号 or 幢. For example: 上海市浦东新区晨晖路 100 弄 10 号 101 室.
Common metro address	Includes the district name, which is common for metropolitan areas in major cities.
Rural address	Includes the village name, which is common for rural addresses.

Parent topic: [Process Chinese addresses \[page 514\]](#)

Related Information

[Sample Chinese address \[page 518\]](#)

[Common metro address components \[page 517\]](#)

[Chinese rural address components \[page 517\]](#)

18.6.5.9.1.1 Common metro address components

Metro addresses in China include the District name, which is common for metropolitan areas in major cities.

The following table contains the Chinese metro address components, Chinese and English translation, and the corresponding output field to use in the transform.

Address component	Chinese	English	Output field
Postcode	510030	510030	<i>Postcode_Full</i>
Country	中国	China	<i>Country</i>
Province	广东省	Guangdong Province	<i>Region1_Full</i>
City name	广州市	Guangzhou City	<i>Locality1_Full</i>
District name	越秀区	Yuexiu District	<i>Locality2_Full</i>
Street name	西湖路	Xihu Road	<i>Primary_Name_Full1</i>
House number	99 号	No. 99	<i>Primary_Number_Full</i>

18.6.5.9.1.2 Chinese rural address components

Chinese rural addresses include the village name as well as the town name and the county-level city name.

The following table lists the Chinese rural address components, Chinese and English translation, and the corresponding output field to use in the transform.

Address component	Chinese	English	Output field
Postcode	5111316	5111316	<i>Postcode_Full</i>
Country	中国	China	<i>Country</i>
Province	广东省	Guangdong Province	<i>Region1_Full</i>
City name	广州市	Guangzhou City	<i>Locality1_Full</i>

Address component	Chinese	English	Output field
County-level City name	增城市	Zengcheng City	<i>Locality2_Full</i>
Town name	荔城镇	Licheng Town	<i>Locality3_Full</i>
Village name	联益村	Lianyi Village	<i>Locality4_Full</i>
Street name	光大路	Guangda Road	<i>Primary_Name_Full1</i>
House number	99 号	No. 99	<i>Primary_Number_Full</i>

18.6.5.9.2 Sample Chinese address

The Global Address Cleanse transform processes the various parts of a Chinese address by parsing the address elements into address line fields, lastline fields, and nonparsed fields.

❖ Example

The following input address is processed by the Global Address Cleanse transform. The transform parses address elements to address-line fields, lastline fields, and nonparsed fields.

Input address

510830 广东省广州市花都区赤坭镇广源路 1 号星辰大厦 8 层 809 室

Address-Line fields

<i>Primary_Name1</i>	广源
<i>Primary_Type1</i>	路
<i>Primary_Number</i>	1
<i>Primary_Number_Description</i>	号
<i>Building_Name1</i>	星辰大厦
<i>Floor_Number</i>	8
<i>Floor_Description</i>	层
<i>Unit_Number</i>	809
<i>Unit_Description</i>	室
<i>Primary_Address</i>	广源路 1 号
<i>Secondary_Address</i>	星辰大厦 8 层 809 室
<i>Primary_Secondary_Address</i>	广源路 1 号星辰大厦 8 层 809 室

Lastline fields

<i>Country</i>	中国

Lastline fields

<i>Postcode_Full</i>	510168
<i>Region1</i>	广东
<i>Region1_Description</i>	省
<i>Locality1_Name</i>	广州
<i>Locality1_Description</i>	市
<i>Locality2_Name</i>	花都
<i>Locality2_Description</i>	区
<i>Locality3_Name</i>	赤坭
<i>Locality3_Description</i>	镇
<i>Lastline</i>	510830 广东省广州市花都区赤坭镇

Non-parsed fields

<i>Status_Code</i>	S0000
<i>Assignment_Type</i>	S
<i>Address_Type</i>	S

Parent topic: [Process Chinese addresses \[page 514\]](#)

Related Information

[Chinese address format \[page 514\]](#)

18.6.5.10 Supported countries (Global Address Cleanse)

There are several countries supported by the Global Address Cleanse transform. The level of correction varies by country and by the engine that you use. Complete coverage of all addresses in a country is not guaranteed.

For the Global Address engine, country support depends on which sets of postal directories you have purchased.

For Japan, the assignment level is based on data provided by the Ministry of Public Management Home Affairs, Posts and Telecommunications (MPT).

During Country ID processing, the transform can identify many countries. However, the Global Address Cleanse transform's engines may not provide address correction for all of those countries.

[Changing the default output country name \[page 520\]](#)

[Setting a default country \[page 521\]](#)

[Setting a constant country \[page 521\]](#)

Parent topic: [Address Cleanse \[page 483\]](#)

Related Information

[How address cleanse works \[page 484\]](#)

[Address cleanse reports \[page 489\]](#)

[Preparing your input data \[page 490\]](#)

[Determining which transform\(s\) to use \[page 492\]](#)

[Identifying the country of destination \[page 497\]](#)

[Set up the reference files \[page 498\]](#)

[Defining the standardization options \[page 501\]](#)

[Process Japanese addresses \[page 503\]](#)

[Process Chinese addresses \[page 514\]](#)

[New Zealand certification \[page 522\]](#)

[Global Address Cleanse Suggestion List option \[page 527\]](#)

[Global Suggestion List transform \[page 528\]](#)

[Address assignment for buildings \[page 530\]](#)

[Changing the default output country name \[page 520\]](#)

[Reference Guide: Country ISO codes and assignment engines](#)

18.6.5.10.1 Changing the default output country name

When you use the USA engine to process addresses from American Samoa, Guam, Northern Mariana Islands, Palau, Puerto Rico, and the U.S. Virgin Islands, the output region is AS, GU, MP, PW, PR, or VI, respectively. The output country, however, is the United States (US).

If you do not want the output country as the United States when processing addresses with the USA engine, set the *Use Postal Country Name* option to *No*.

These steps show you how to set the Use Postal Country Name in the Global Address Cleanse transform.

1. Open the *Global Address Cleanse* transform.
2. On the *Options* tab, expand  *Standardization Options*  *Country*  *Options* .
3. For the *Use Postal Country Name* option, select *No*.

Task overview: [Supported countries \(Global Address Cleanse\) \[page 519\]](#)

Related Information

[Setting a default country \[page 521\]](#)

[Setting a constant country \[page 521\]](#)

[Supported countries \(Global Address Cleanse\) \[page 519\]](#)

18.6.5.10.2 Setting a default country

i Note

Run Country ID processing only if you are:

- Using two or more of the engines and your input addresses contain country data (such as the two-character ISO code or a country name).
- Using only one engine, but your input data contains addresses from multiple countries.

1. Open the [Global Address Cleanse](#) transform.
2. On the [Options](#) tab, expand [Country ID Options](#), and then for the [Country ID Mode](#) option select [Assign](#). This value directs the transform to use [Country ID](#) to assign the country. If [Country ID](#) cannot assign the country, it will use the value in [Country Name](#).
3. For the [Country Name](#) option, select the country that you want to use as a default country.

The transform will use this country only when [Country ID](#) cannot assign a country. If you do not want a default country, select [None](#).
4. For the [Script Code](#) option, select the type of script code that represents your data.

The [LATN](#) option provides script code for most types of data. However, if you are processing Japanese data, select [KANJI](#)

Task overview: [Supported countries \(Global Address Cleanse\) \[page 519\]](#)

Related Information

[Changing the default output country name \[page 520\]](#)

[Setting a constant country \[page 521\]](#)

[Identifying the country of destination \[page 497\]](#)

[Setting a constant country \[page 521\]](#)

18.6.5.10.3 Setting a constant country

1. Open the [Global Address Cleanse](#) transform.

2. On the [Options](#) tab, expand [Country ID Options](#), and then for the [Country ID Mode](#) option select [Constant](#). This value tells the transform to take the country information from the [Country Name](#) and [Script Code](#) options (instead of running Country ID processing).
3. For the [Country Name](#) option, select the country that represents all your input data.
4. For the [Script Code](#) option, select the type of script code that represents your data.
The [LATN](#) option provides script code for most types of data. However, if you are processing Japanese data, select [KANJI](#)

Task overview: [Supported countries \(Global Address Cleanse\)](#) [page 519]

Related Information

[Changing the default output country name](#) [page 520]

[Setting a default country](#) [page 521]

[Identifying the country of destination](#) [page 497]

[Setting a default country](#) [page 521]

18.6.5.11 New Zealand certification

New Zealand certification enables you to process New Zealand addresses and qualify for mailing discounts with the New Zealand Post.

[Enabling New Zealand certification](#) [page 523]

[New Zealand Statement of Accuracy Report](#) [page 524]

[New Zealand SOA Production Log](#) [page 524]

The Production Log is a pipe-delimited ASCII text file (with a header record).

[Mailing requirements](#) [page 525]

The SOA report and production log are only required when you submit the data processed for a mailing and want to receive postage discounts.

[Editing the New Zealand certification blueprint](#) [page 525]

[Editing the datastore](#) [page 526]

Parent topic: [Address Cleanse](#) [page 483]

Related Information

[How address cleanse works](#) [page 484]

[Address cleanse reports](#) [page 489]

[Preparing your input data](#) [page 490]

[Determining which transform\(s\) to use \[page 492\]](#)
[Identifying the country of destination \[page 497\]](#)
[Set up the reference files \[page 498\]](#)
[Defining the standardization options \[page 501\]](#)
[Process Japanese addresses \[page 503\]](#)
[Process Chinese addresses \[page 514\]](#)
[Supported countries \(Global Address Cleanse\) \[page 519\]](#)
[Global Address Cleanse Suggestion List option \[page 527\]](#)
[Global Suggestion List transform \[page 528\]](#)
[Address assignment for buildings \[page 530\]](#)

18.6.5.11.1 Enabling New Zealand certification

You need to purchase the New Zealand directory data and obtain a customer number from the New Zealand Post before you can use the New Zealand certification option.

To process New Zealand addresses that qualify for mailing discounts:

1. In the Global Address Cleanse Transform, enable ► [Report and Analysis](#) ► [Generate Report Data](#) ►.
2. In the Global Address Cleanse Transform, set ► [Country Options](#) ► [Disable Certification](#) ► to *No*.

i Note

The software does not produce the New Zealand Statement of Accuracy (SOA) report when this option is set to *Yes*.

3. In the Global Address Transform, complete all applicable options in the ► [Global Address](#) ► [Report Options](#) ► [New Zealand](#) ► subgroup.
4. In the Global Address Cleanse Transform, set ► [Engines](#) ► [Global Address](#) ► to *Yes*.

After you run the job and produce the New Zealand Statement of Accuracy (SOA) report, you need to rename the New Zealand Statement of Accuracy (SOA) report and New Zealand Statement of Accuracy (SOA) Production Log before submitting your mailing.

Task overview: [New Zealand certification \[page 522\]](#)

Related Information

[New Zealand Statement of Accuracy Report \[page 524\]](#)
[New Zealand SOA Production Log \[page 524\]](#)
[Mailing requirements \[page 525\]](#)
[Editing the New Zealand certification blueprint \[page 525\]](#)
[Editing the datastore \[page 526\]](#)
[New Zealand Statement of Accuracy Report \[page 524\]](#)

[Management Console Guide: New Zealand Statement of Accuracy \(SOA\) report](#)
[Reference Guide: Report options for New Zealand](#)

18.6.5.11.2 New Zealand Statement of Accuracy Report

The New Zealand Statement of Accuracy (SOA) report includes statistical information about address cleansing for New Zealand.

Parent topic: [New Zealand certification \[page 522\]](#)

Related Information

[Enabling New Zealand certification \[page 523\]](#)
[New Zealand SOA Production Log \[page 524\]](#)
[Mailing requirements \[page 525\]](#)
[Editing the New Zealand certification blueprint \[page 525\]](#)
[Editing the datastore \[page 526\]](#)
[Management Console Guide: New Zealand Statement of Accuracy \(SOA\) report](#)
[Management Console Guide: Exporting New Zealand SOA certification logs](#)
[Reference Guide: Report options for New Zealand](#)

18.6.5.11.3 New Zealand SOA Production Log

The Production Log is a pipe-delimited ASCII text file (with a header record).

The Production Log is identical to the New Zealand Statement of Accuracy (SOA) report except it is in a pipe-delimited ASCII text file (with a header record). The software creates the SOA Production Log by extracting data from the Sendrightaddraccuracy table within the repository. The software appends a new record to the Sendrightaddraccuracy table each time a file is processed with the DISABLE_CERTIFICATION option set to *No*. If the DISABLE_CERTIFICATION option is set to *Yes*, the software does not produce the SOA report and an entry will not be appended to the Sendrightaddraccuracy table.

Mailers **must** retain the production log file for at least 2 years.

The default location of the SOA production log is `<INSTALL_DIR>\Data Services\DataQuality\certifications\CertificationLogs`.

Parent topic: [New Zealand certification \[page 522\]](#)

Related Information

[Enabling New Zealand certification \[page 523\]](#)

[New Zealand Statement of Accuracy Report \[page 524\]](#)

[Mailing requirements \[page 525\]](#)

[Editing the New Zealand certification blueprint \[page 525\]](#)

[Editing the datastore \[page 526\]](#)

18.6.5.11.4 Mailing requirements

The SOA report and production log are only required when you submit the data processed for a mailing and want to receive postage discounts.

- Submit the SOA production log at least once a month.
- Submit an SOA report for each file that is processed for mailing discounts.

Parent topic: [New Zealand certification \[page 522\]](#)

Related Information

[Enabling New Zealand certification \[page 523\]](#)

[New Zealand Statement of Accuracy Report \[page 524\]](#)

[New Zealand SOA Production Log \[page 524\]](#)

[Editing the New Zealand certification blueprint \[page 525\]](#)

[Editing the datastore \[page 526\]](#)

[New Zealand certification \[page 522\]](#)

[New Zealand Statement of Accuracy Report \[page 524\]](#)

[New Zealand SOA Production Log \[page 524\]](#)

18.6.5.11.5 Editing the New Zealand certification blueprint

Follow the steps below to edit the New Zealand certification blueprint:

1. Import the file `nz_sendright_certification.atl` located in the `DataQuality\certifications` folder where you installed the software.

The default location is `<INSTALL_DIR>\Data Services\DataQuality\certifications`.

The import adds the following objects to the repository:

- The project `DataQualityCertifications`
- The job `Job_DqBatchNewZealand_SOAProductionLog`

- The data flow DF_DqBatchNewZealand_SOAProductionLog
 - The datastore DataQualityCertifications
 - The file format DqNewZealandSOAProductionLog
2. Edit the datastore DataQualityCertifications object by following the steps in [Editing the datastore \[page 526\]](#).
 3. Keep the default location for the SOA Production log or change it to a different location by editing the substitution parameter configuration:
 - a. From the Designer, select **Tools** > **Substitution Parameter Configurations**.
 - b. Change the path location in Configuration1 for the substitution parameter \$CertificationLogPath to the location of your choice.

The default location for the SOA Production Log is <INSTALL_DIR>\Data Services\DataQuality\certifications\CertificationLogs.
 4. Run the job Job_DqBatchNewZealand_SOAProductionLog.
The job produces an SOA Production Log called SOAPerc_SOAExpDate_SOAIId.txt in the default location or the location you specified in the substitution parameter configuration.
 5. Rename the SOAPerc_SOAExpDate_SOAIId.txt file using data from the last record in the log file and the file naming format described in [New Zealand Statement of Accuracy Report \[page 524\]](#).

Task overview: [New Zealand certification \[page 522\]](#)

Related Information

[Enabling New Zealand certification \[page 523\]](#)

[New Zealand Statement of Accuracy Report \[page 524\]](#)

[New Zealand SOA Production Log \[page 524\]](#)

[Mailing requirements \[page 525\]](#)

[Editing the datastore \[page 526\]](#)

[New Zealand Statement of Accuracy Report \[page 524\]](#)

[Management Console Guide: New Zealand Statement of Accuracy \(SOA\) report](#)

18.6.5.11.6 Editing the datastore

After you download the blueprint .zip file to the appropriate folder, unzip it, and import the .atl file in the software, you must edit the DataQualityCertifications datastore.

To edit the datastore:

1. Select the **Datastores** tab of the Local Object Library, right-click DataQualityCertifications and select **Edit**.
2. Click **Advanced** to expand the Edit Datastore DataQualityCertifications window.

i Note

Skip step 3 if you have Microsoft SQL Server 2000 or 2005 as a datastore database type.

3. Click [Edit](#).
4. Find the column for your database type, change [Default configuration](#) to **Yes**, and click [OK](#).

i Note

If you are using a version of Oracle other than Oracle 9i, perform the following substeps:

- a. In the toolbar, click [Create New Configuration](#).
 - b. Enter your information, including the Oracle database version that you are using, and then click [OK](#).
 - c. Click [Close](#) on the Added New Values - Modified Objects window.
 - d. In the new column that appears to the right of the previous columns, select [Yes](#) for the [Default configuration](#).
 - e. Enter your information for the [Database connection name](#), [User name](#), and [Password](#) options.
 - f. In [DBO](#), enter your schema name.
 - g. In [Code Page](#), select [cp1252](#) and then click [OK](#).
5. At the Edit Datastore DataQualityCertifications window, enter your repository connection information in place of the CHANGE_THIS values. (You may have to change three or four options, depending on your repository type.)
 6. Expand the [Aliases](#) group and enter your owner name in place of the CHANGE_THIS value. If you are using Microsoft SQL Server, set this value to **DBO**.
 7. Click [OK](#).
- If the window closes without any error message, then the database is successfully connected.

Task overview: [New Zealand certification \[page 522\]](#)

Related Information

[Enabling New Zealand certification \[page 523\]](#)

[New Zealand Statement of Accuracy Report \[page 524\]](#)

[New Zealand SOA Production Log \[page 524\]](#)

[Mailing requirements \[page 525\]](#)

[Editing the New Zealand certification blueprint \[page 525\]](#)

18.6.5.12 Global Address Cleanse Suggestion List option

The Global Address Cleanse transform's Suggestion List processing option is used in transactional projects to complete and populate addresses that have minimal data.

Suggestion lists can offer suggestions for possible matches if an exact match is not found. The suggestion list feature is beneficial in situations where you want to extract addresses not completely assigned by an automated process, and run through the system to find a list of possible matches. Based on the given input address, the Global Address Cleanse transform performs an error-tolerant search in the address directory and returns a list of possible matches. From the suggestion list returned, you can select the correct suggestion and update the database accordingly.

i Note

No certification with suggestion lists: If you use the Canada engine or Global Address engine for Australia and New Zealand, you cannot certify your mailing for SERP, AMAS, or New Zealand certification.

Start with a sample transform

If you want to use the suggestion lists feature, it is best to start with the sample transforms that are configured for suggestion lists. The sample transform, `GlobalSuggestions_AddressCleanse` is configured to cleanse Latin-1 address data in any supported country using the Suggestion List feature.

Parent topic: [Address Cleanse \[page 483\]](#)

Related Information

[How address cleanse works \[page 484\]](#)

[Address cleanse reports \[page 489\]](#)

[Preparing your input data \[page 490\]](#)

[Determining which transform\(s\) to use \[page 492\]](#)

[Identifying the country of destination \[page 497\]](#)

[Set up the reference files \[page 498\]](#)

[Defining the standardization options \[page 501\]](#)

[Process Japanese addresses \[page 503\]](#)

[Process Chinese addresses \[page 514\]](#)

[Supported countries \(Global Address Cleanse\) \[page 519\]](#)

[New Zealand certification \[page 522\]](#)

[Global Suggestion List transform \[page 528\]](#)

[Address assignment for buildings \[page 530\]](#)

[Extract data quality XML strings using `extract\_from\_xml` function \[page 688\]](#)

[Suggestion List: Global Address Cleanse](#)

18.6.5.13 Global Suggestion List transform

The Global Suggestion List transform allows you to query addresses with minimal data (allowing the use of wildcards), and it can offer a list of suggestions for possible matches.

It is a beneficial tool for a call center environment, where operators need to enter minimum input (number of keystrokes) to find the caller's delivery address. For example, if the operator is on the phone with a caller from the United Kingdom, the application prompts for the postcode and address range. Global Suggestion List is used to look up the address with quick-entry.

The Global Suggestion List transform requires the two-character ISO country code on input. Therefore, you may want to place a transform, such as the Country ID transform, that outputs the ISO_Country_Code_2Char field before the Global Suggestion Lists transform. The Global Suggestion List transform is available for use with the Canada, Global Address, and USA engines.

i Note

No certification with suggestion lists: If you use the Canada, Global Address, or USA engines for Australia and New Zealand, you cannot certify your mailing for SERP, CASS, AMAS, or New Zealand certification.

i Note

This option does not support processing of Japanese or Chinese address data.

Start with a sample transform

If you want to use the Global Suggestion List transform, it is best to start with one of the sample transforms that is configured for it. The following sample transforms are available.

Sample transform	Description
GlobalSuggestions	A sample transform configured to generate a suggestion list for Latin-1 address data in any supported country.
UKSuggestions	A sample transform configured to generate a suggestion list for partial address data in the United Kingdom.

Parent topic: [Address Cleanse \[page 483\]](#)

Related Information

[How address cleanse works \[page 484\]](#)
[Address cleanse reports \[page 489\]](#)
[Preparing your input data \[page 490\]](#)
[Determining which transform\(s\) to use \[page 492\]](#)
[Identifying the country of destination \[page 497\]](#)
[Set up the reference files \[page 498\]](#)
[Defining the standardization options \[page 501\]](#)
[Process Japanese addresses \[page 503\]](#)
[Process Chinese addresses \[page 514\]](#)
[Supported countries \(Global Address Cleanse\) \[page 519\]](#)
[New Zealand certification \[page 522\]](#)
[Global Address Cleanse Suggestion List option \[page 527\]](#)
[Address assignment for buildings \[page 530\]](#)

18.6.5.14 Address assignment for buildings

The Global Address Cleanse transform uses building information in addresses to enhance the address assignment process and results.

Addresses from various countries contain building information. For example, an address can contain a building name, building descriptor, building number, or building type. Additionally, some countries separate the base building name and the building descriptor in reference data, so the Global Address Cleanse transform accommodates the information in specific building output fields.

The Global Address Cleanse transform uses building information to make address assignments. For example, an input address may have building information, but no street address, or a partial street address and a building number.

When your data contains addresses with building information, configure the Global Address Cleanse transform to include output fields for building information.

Output fields for parsed building information

Output field	Building information
<i>BUILDING1</i> <i>BUILDING2</i>	Contains the base name for a building. The transform separates the base name from the description for countries that separate the two in reference data. ❖ Example A building name in the Russian Federation is “корпус Б.” The transform outputs building data as follows: • <i>BUILDING1</i> contains the base name “Б” • <i>BUILDING_DESCRIPTION1</i> contains the descriptor “корпус” The <i>BUILDING2</i> field contains the base name for the second building in an address when there are two buildings in an address.

Output field	Building information
<i>BUILDING_DESCRIPTION1</i> <i>BUILDING_DESCRIPTION2</i>	Contains the building descriptor. For countries that separate the base building name from the descriptor in reference data, the transform outputs the building description to this field. ❖ Example The transform outputs the building name "корпус Б" from the Russian Federation as follows: • <i>BUILDING_DESCRIPTION1</i> contains the descriptor "корпус" • <i>BUILDING1</i> contains the base name "Б" <i>BUILDING_DESCRIPTION2</i> contains the building descriptor for the second building in an address that contains two buildings.
<i>BUILDING_NAME1</i> <i>BUILDING_NAME2</i>	Contains the full building name, which includes the building name and the building descriptor. ❖ Example "White House" or "Empire Tower" <i>BUILDING_NAME2</i> contains the second building name for addresses with two buildings.
<i>ASSIGNMENT_LEVEL</i>	An input address contains the building name and a last line with other secondary information. Data Services makes an address assignment using the secondary information. It may output an <i>S</i> in the <i>ASSIGNMENT_LEVEL</i> output field because it made an assignment based on secondary information. An input address contains a primary street number, building information, and a lastline. Data Services makes an address assignment based on the primary street number. It may output a <i>PR</i> in the <i>ASSIGNMENT_LEVEL</i> output field because it made an assignment based on the primary number.
<i>PRIMARY_NUMBER_FULL</i>	When an input address includes a primary name and number, Data Services outputs the primary number without a descriptor. However, when there is not a primary name on input, Data Services outputs the primary number and descriptor to the <i>PRIMARY_NUMBER_FULL</i> output field. Data Services also outputs the primary number and descriptor to the <i>PRIMARY_NUMBER_FULL</i> field when the input address contains a building and primary number, or only a primary number.

Swap or combine building numbers

When an input has two numbers in a row in the address line, Data Services may combine the numbers and attempt to assign a building number. When an input address has two numbers in a row separated by a slash, period, or other delimiter, the transform may swap the number order and then combine the number to make an address assignment.

Standardize building descriptor

Use standardization options in the Standardize Group to control how the software outputs address components, including building information. The following list contains some settings you can make to control the format of your output data:

- *Capitalization:*
 - Set the option to *MIXED* to output the descriptor to the *BUILDING_DESCRIPTION1* output field in lower case with the first letter capitalized.
 - Set the option to *UPPER* to output the descriptor to the *BUILDING_DESCRIPTION1* output field in all upper case letters.
- *Primary Type Style:*
 - Set the output style of the primary number descriptor to *LONG*, *SHORT*, or *PRESERVE*. The *Primary Type Style* setting affects the building descriptor, for example, in the *BUILDING_NAME* output field. The *BUILDING_NAME* output field contains the full building name, which includes the building name and the building descriptor.
- *Secondary Description Style:*
 - Set the output style of the *BUILDING_DESCRIPTION1* or *BUILDING_DESCRIPTION2* to output in long form, abbreviated form, or preserved to output the descriptor as it was input.

For complete field descriptions, see the *Reference Guide*.

Parent topic: [Address Cleanse \[page 483\]](#)

Related Information

[How address cleanse works \[page 484\]](#)

[Address cleanse reports \[page 489\]](#)

[Preparing your input data \[page 490\]](#)

[Determining which transform\(s\) to use \[page 492\]](#)

[Identifying the country of destination \[page 497\]](#)

[Set up the reference files \[page 498\]](#)

[Defining the standardization options \[page 501\]](#)

[Process Japanese addresses \[page 503\]](#)

[Process Chinese addresses \[page 514\]](#)

[Supported countries \(Global Address Cleanse\) \[page 519\]](#)

[New Zealand certification \[page 522\]](#)

[Global Address Cleanse Suggestion List option \[page 527\]](#)

[Global Suggestion List transform \[page 528\]](#)

18.6.6 Beyond the basic address cleansing

The USA Regulatory Address Cleanse transform offers many additional address cleanse features for U.S. addresses. These features extend address cleansing beyond the basic parsing and standardizing. To read about the USA Regulatory Address Cleanse transform and its options, see the *Reference Guide*.

[USPS DPV \[page 534\]](#)

DPV (Delivery Point Validation) is a USPS product that assists users in validating the accuracy of their address information.

[LACSLink \[page 557\]](#)

LACSLink updates addresses when the physical address does not move but the address is changed.

[SuiteLink \[page 567\]](#)

Use the SuiteLink directory information to add suite numbers and other secondary data to high-rise business addresses.

[USPS DSF2® \[page 570\]](#)

[NCOALink and Move Update \[page 581\]](#)

National Change of Address (NCOALink) is a part of USPS Move Update, which ensures that your mailing list contains up-to-date address information.

[USPS eLOT \[page 598\]](#)

eLOT narrows the mail carrier delivery route walk sequence to the house (delivery point) level.

[Early Warning System \(EWS\) \[page 599\]](#)

EWS helps reduce the amount of misdirected mail caused when the USPS adds valid delivery points between national directory updates.

[USPS RDI \[page 601\]](#)

RDI (Residential Delivery Indicator) determines whether a given address is for a residence or non-residence.

[GeoCensus \(USA Regulatory Address Cleanse \) \[page 604\]](#)

[Z4Change \[page 608\]](#)

The Z4Change feature enhances automation compatibility and improves deliverability.

[Suggestion lists \[page 610\]](#)

Use Suggestion List processing in transactional projects with the USA Regulatory Address Cleanse, Global Address Cleanse, and the Global Suggestion List transforms.

[Multiple data source statistics reporting \[page 613\]](#)

The USA Regulatory Address Cleanse transform reports statistics for a job that contains multiple data sources based on logical groupings.

18.6.6.1 USPS DPV

DPV (Delivery Point Validation) is a USPS product that assists users in validating the accuracy of their address information.

DPV compares Postcode2 information against the DPV directories to identify known addresses and potential problems with the address that may cause an address to become undeliverable.

DPV is available for U.S. data in the USA Regulatory Address Cleanse transform only.

i Note

DPV processing is required for CASS certification. If you are not processing for CASS certification, you can choose to run your jobs in non-certified mode and still enable DPV.

⚠ Caution

If you choose to disable DPV processing, the software does not generate the CASS-required documentation and your mailing is not eligible for postal discounts.

[Benefits of DPV \[page 535\]](#)

Take advantage of the many benefits of enabling the Delivery Point Validation (DPV) product.

[DPV security \[page 536\]](#)

To ensure that the DPV data is not misused, the USPS has processes that monitor the use of DPV.

[DPV monthly directories \[page 542\]](#)

SAP ships the DPV directories monthly along with the USPS directories according to USPS guidelines.

[Required information in the job setup \[page 543\]](#)

Complete the USPS-required components in the transform configuration.

[Enabling DPV \[page 544\]](#)

[DPV output fields \[page 545\]](#)

To obtain DPV output information from your job processing, include specific output fields in your output file setup.

[Non-certified mode \[page 549\]](#)

Assign Postcode2 data to addresses that do not DPV confirm.

[DPV performance \[page 550\]](#)

The additional time required to perform DPV processing may increase processing time.

[DPV information in US Addressing Report \[page 552\]](#)

[DPV No Stats indicators \[page 553\]](#)

The USPS designates addresses as No Stats when the address is a vacant property, it receives mail as part of a drop, or it doesn't have an established delivery yet.

[DPV Vacant indicators \[page 554\]](#)

The USPS assigns a DPV Vacant indicator to addresses that had an active DPV but now it is vacant (for over 90 days), it doesn't receive mail delivery, and it can receive mail delivery in the future.

[DPV PO Box Throwback indicator \[page 555\]](#)

The USPS uses the PO Box Throwback table to identify delivery points that are street addresses but mail delivery is to a PO Box address instead.

[DPV Door Not Accessible indicator \[page 556\]](#)

The USPS uses Door Not Accessible (DNA) indicators to mark addresses where the carrier can't knock on the door for mail delivery.

Parent topic: [Beyond the basic address cleansing \[page 533\]](#)

Related Information

[LACSLink \[page 557\]](#)

[SuiteLink \[page 567\]](#)

[USPS DSF2® \[page 570\]](#)

[NCOALink and Move Update \[page 581\]](#)

[USPS eLOT \[page 598\]](#)

[Early Warning System \(EWS\) \[page 599\]](#)

[USPS RDI \[page 601\]](#)

[GeoCensus \(USA Regulatory Address Cleanse \) \[page 604\]](#)

[Z4Change \[page 608\]](#)

[Suggestion lists \[page 610\]](#)

[Multiple data source statistics reporting \[page 613\]](#)

[Enabling DPV \[page 544\]](#)

[Non-certified mode \[page 549\]](#)

18.6.6.1.1 Benefits of DPV

Take advantage of the many benefits of enabling the Delivery Point Validation (DPV) product.

When you process your address data with DPV enabled, you benefit by reduced mailing costs, enhanced information quality, increased assignment rate, and mail-order fraud prevention:

- **Reduced mailing costs:** DPV screens out undeliverable-as-addressed (UAA) mail, which reduces the number of mail pieces you pay for that do not get delivered.
- **Enhanced information quality:** DPV increases the level of data accuracy by verifying an address down to the individual house, suite, or apartment instead of only the block face.
- **Increased assignment rate:** DPV uses tie breaking to resolve an address tie where other tie-breaking methods are not conclusive. Tie breaking may increase assignment rate.
- **Preventing mail-order-fraud:** DPV can eliminate shipping of merchandise to individuals who place fraudulent orders by verifying valid delivery addresses and Commercial Mail Receiving Agencies (CMRA).

Parent topic: [USPS DPV \[page 534\]](#)

Related Information

[DPV security \[page 536\]](#)
[DPV monthly directories \[page 542\]](#)
[Required information in the job setup \[page 543\]](#)
[Enabling DPV \[page 544\]](#)
[DPV output fields \[page 545\]](#)
[Non-certified mode \[page 549\]](#)
[DPV performance \[page 550\]](#)
[DPV information in US Addressing Report \[page 552\]](#)
[DPV No Stats indicators \[page 553\]](#)
[DPV Vacant indicators \[page 554\]](#)
[DPV PO Box Throwback indicator \[page 555\]](#)
[DPV Door Not Accessible indicator \[page 556\]](#)

18.6.6.1.2 DPV security

To ensure that the DPV data is not misused, the USPS has processes that monitor the use of DPV.

When a company purchases the DPV functionality, the USPS requires them to sign a legal agreement. The agreement states that the company agrees not to misuse the DPV product. If the company abuses the DPV product, the USPS can prohibit the user from using DPV in the future.

[DPV false positive addresses \[page 537\]](#)

The USPS includes false positive addresses in the DPV directories as an added security to prevent DPV data abuse.

[Stop Processing Alternative \[page 538\]](#)

A Stop Processing Alternative agreement enables you to bypass any DPV or LACSLink directory locks.

[DPV false positive logs \[page 539\]](#)

The software generates a false-positive log file any time it encounters a DPV false positive record.

[DPV locking for end users \[page 540\]](#)

If you are an end user without a Stop Processing Alternative agreement with the USPS and the software encounters a false positive record in your database, it stops DPV processing for the remainder of the data flow.

[Obtaining DPV unlock code from SAP \[page 540\]](#)

When the transform locks DPV processing because it has found a false positive address in your data, you unlock DPV processing by obtaining an unlock code from SAP.

[Sending DPV false positive logs to the USPS \[page 542\]](#)

If you receive a message from the software that DPV false positive addresses are present in your address list, obtain a list release from the USPS before you use the list again.

Parent topic: [USPS DPV \[page 534\]](#)

Related Information

[Benefits of DPV \[page 535\]](#)
[DPV monthly directories \[page 542\]](#)
[Required information in the job setup \[page 543\]](#)
[Enabling DPV \[page 544\]](#)
[DPV output fields \[page 545\]](#)
[Non-certified mode \[page 549\]](#)
[DPV performance \[page 550\]](#)
[DPV information in US Addressing Report \[page 552\]](#)
[DPV No Stats indicators \[page 553\]](#)
[DPV Vacant indicators \[page 554\]](#)
[DPV PO Box Throwback indicator \[page 555\]](#)
[DPV Door Not Accessible indicator \[page 556\]](#)

18.6.6.1.2.1 DPV false positive addresses

The USPS includes false positive addresses in the DPV directories as an added security to prevent DPV data abuse.

A false positive address is a fake address that the USPS places in the DPV directories. When the software encounters a false positive address in your data, it can stop DPV processing or continue processing based on your DPV user type. Your action when the software encounters a false positive also depends on your user type.

The following table explains the behaviors for each user type.

User type	Software behavior	Action
End users without a Stop Processing Alternative agreement	DPV directories lock and DPV processing terminates. Job continues to run without DPV.	1. Unlock DPV processing by obtaining a DPV unlock code from SAP Support. 2. Send false positive logs to the USPS to release the mailing list.
End user with Stop Processing Alternative agreement	DPV processing continues.	Send false positive logs to the USPS to release the mailing list
Service providers	DPV processing continues.	Send false positive logs to the USPS to release the mailing list.

Parent topic: [DPV security \[page 536\]](#)

Related Information

[Stop Processing Alternative \[page 538\]](#)

[DPV false positive logs \[page 539\]](#)
[DPV locking for end users \[page 540\]](#)
[Obtaining DPV unlock code from SAP \[page 540\]](#)
[Sending DPV false positive logs to the USPS \[page 542\]](#)
[Stop Processing Alternative \[page 538\]](#)
[Obtaining DPV unlock code from SAP \[page 540\]](#)
[Sending DPV false positive logs to the USPS \[page 542\]](#)

18.6.6.1.2.2 Stop Processing Alternative

A Stop Processing Alternative agreement enables you to bypass any DPV or LACSLink directory locks.

If you are a DPV or LACSLink end user, you may establish a Stop Processing Alternative agreement with the USPS and SAP.

The Stop Processing Alternative is not an option in the software, it is a key code that you obtain from SAP Support.

Before SAP issues the key code, obtain the proper permissions from the USPS. Then, provide the proof of permission to SAP Customer Support. Customer Support provides a key code that disables the DPV or LACSLink directory lock function in the software.

With the Stop Processing Alternative key code in place, the software takes the following actions when a false positive is encountered:

- Marks the record as a false positive
- Generates a log file containing the false positive address
- Notes the path to the log files in the error log
- Generates a US Regulatory Locking Report containing the path to the log file
- Continues processing your job

The Stop Processing Alternative saves you time. The DPV or LACSLink processing continues so you do not have to follow the process to unlock the directories and rerun the job. However, you still have to send the false positive log to the USPS to notify them that your database contained a false positive record. The USPS has to release the list before you can use the database for production.

Parent topic: [DPV security \[page 536\]](#)

Related Information

[DPV false positive addresses \[page 537\]](#)
[DPV false positive logs \[page 539\]](#)
[DPV locking for end users \[page 540\]](#)
[Obtaining DPV unlock code from SAP \[page 540\]](#)
[Sending DPV false positive logs to the USPS \[page 542\]](#)
[Sending DPV false positive logs to the USPS \[page 542\]](#)

18.6.6.1.2.3 DPV false positive logs

The software generates a false-positive log file any time it encounters a DPV false positive record.

The software creates a separate log file for each mailing list that contains a false positive record. If multiple false positives exist within one mailing list, the software writes them all to the same log file.

The software stores DPV false positive log files in the directory that you specify. Set the DPV false positive log file location in the [USPS_Log_Path](#) option in the Reference Files group of the transform configuration.

i Note

Ensure that the directory that you set in [USPS_Log_Path](#) is writable. The software issues an error if the directory is not writable.

The software uses the following naming convention when it generates a DPV false positive log: `dpvl####.log`, where `####` is a number between 0001 and 9999.

❖ Example

For the first database in which the software finds a false positive record, it names the log file `dpv10001.log`. For the next database in which the software finds a false positive record, it names the log file `dpv10002.log`.

i Note

When you have set the degree of parallelism (DOP) to greater than 1, the software generates 1 log per thread that contains a false positive.

Parent topic: [DPV security \[page 536\]](#)

Related Information

[DPV false positive addresses \[page 537\]](#)

[Stop Processing Alternative \[page 538\]](#)

[DPV locking for end users \[page 540\]](#)

[Obtaining DPV unlock code from SAP \[page 540\]](#)

[Sending DPV false positive logs to the USPS \[page 542\]](#)

18.6.6.1.2.4 DPV locking for end users

If you are an end user without a Stop Processing Alternative agreement with the USPS and the software encounters a false positive record in your database, it stops DPV processing for the remainder of the data flow.

In addition to terminating DPV processing, the software also takes the following actions:

- Marks the record as a false positive address.
- Issues a message in the error log stating that a DPV false positive address was encountered.
- Includes the false positive address and lock code in the error log.
- Continues processing your data flow without DPV processing.
- Generates a lock code.
- Generates a false positive log.
- Generates a US Regulatory Locking Report that contains the false positive address and the lock code. (Report generation must be enabled in the USA Regulatory Address Cleanse transform.)

To restore DPV functionality in the software, obtain a DPV unlock code from SAP customer support.

Parent topic: [DPV security \[page 536\]](#)

Related Information

[DPV false positive addresses \[page 537\]](#)

[Stop Processing Alternative \[page 538\]](#)

[DPV false positive logs \[page 539\]](#)

[Obtaining DPV unlock code from SAP \[page 540\]](#)

[Sending DPV false positive logs to the USPS \[page 542\]](#)

[Obtaining DPV unlock code from SAP \[page 540\]](#)

18.6.6.1.2.5 Obtaining DPV unlock code from SAP

When the transform locks DPV processing because it has found a false positive address in your data, you unlock DPV processing by obtaining an unlock code from SAP.

The following steps apply to end users who do not have a Stop Processing Alternative agreement with the USPS.

Before you perform the following steps, write down the lock code and the locking address from the `dpvx.txt` file. The file is located in the directory you specify in the *DPV Path* option in the Reference Files settings of the transform.

1. Go to <https://support.sap.com/en/my-support/keys.html> and scroll down to United States Postal Services (USPS) Unlock Code section.
2. Click [Get your USPS Unlock Codes](#).
3. Click [Search](#) and select an applicable Data Services system from the list.

4. Enter the lock code that you wrote down from the `dpvx.txt` file.
5. Select DPV as the lock type.
6. Select BOJ-EIM-DS as the component.
7. Enter the address that caused the lock that you wrote down from the `dpvx.txt` file.
8. Attach the `dpv1####.log` log file.

You specify the location in the [USPS Log Path](#) option in the Reference Files group.

9. Click [Submit](#).

The unlock code displays.

i Note

If the unlock code could not be generated, the system sends a message to a Technical Customer Assurance engineer who generates an unlock code for you during regular business hours.

10. Open the `dpvw.txt` file, delete all content, and paste the unlock code into the file.

i Note

The `dpvw.txt` file is different than the `dpvx.txt` file.

11. Remove the record that caused the lock from the database
12. Delete the applicable `dpv1####.log` file.

You can use the unlock code only one time. If the software detects another false-positive, even if it is the same record, you have to retrieve a new DPV unlock code. Therefore, before you process the list again, make sure that you perform steps 11 and 12.

All DPV users must send the false positive log to the USPS to obtain release of the mailing list. Follow the steps in [Sending DPV false positive logs to the USPS \[page 542\]](#).

Task overview: [DPV security \[page 536\]](#)

Related Information

[DPV false positive addresses \[page 537\]](#)

[Stop Processing Alternative \[page 538\]](#)

[DPV false positive logs \[page 539\]](#)

[DPV locking for end users \[page 540\]](#)

[Sending DPV false positive logs to the USPS \[page 542\]](#)

18.6.6.1.2.6 Sending DPV false positive logs to the USPS

If you receive a message from the software that DPV false positive addresses are present in your address list, obtain a list release from the USPS before you use the list again.

This process applies to all DPV user types, regardless of whether they have a Stop Processing Alternative agreement with the USPS.

1. Send an email to the USPS NCSC (National Customer Support Center) at dsf2stop@usps.gov, and include the following information:
 - Enter the following in the e-mail subject line: "DPV False Positive"
 - Attach the `dpv1####.log` file or files

The USPS NCSC uses the information submitted with the e-mail to determine whether it can release the list to the mailer.

2. After the USPS NCSC has released the list that contained the locked or false positive record, perform the following tasks:
 1. Delete the corresponding log file or files (`dpv1####.log`).
 2. Delete the false positive record from your database.
3. Reprocess the database.

End users who do not have a Stop Processing Alternative agreement can obtain a DPV unlock code from SAP by following the steps in [Obtaining DPV unlock code from SAP \[page 540\]](#)

Task overview: [DPV security \[page 536\]](#)

Related Information

[DPV false positive addresses \[page 537\]](#)

[Stop Processing Alternative \[page 538\]](#)

[DPV false positive logs \[page 539\]](#)

[DPV locking for end users \[page 540\]](#)

[Obtaining DPV unlock code from SAP \[page 540\]](#)

18.6.6.1.3 DPV monthly directories

SAP ships the DPV directories monthly along with the USPS directories according to USPS guidelines.

The directories expire in 105 days (four months). Ensure that the DPV directories and the Address directory files have the same date. Do not process files with expired directories.

Do not rename any of the DPV files. If you change the file names, DPV processing does not run. The following list contains the DPV directory files:

- `dpva.dir`

- `dpvb.dir`
- `dpvc.dir`
- `dpvd.dir`
- `dpv_door_na.dir`
- `dpv_vacant.dir`
- `dpv_no_stats.dir`
- `dpv_non_delivery_days_flag.dir`
- `dpv_non_delivery_days.dir`
- `dpv_po_box_throwback.dir`
- `dpv_no_secure_loc.dir`

Parent topic: [USPS DPV \[page 534\]](#)

Related Information

[Benefits of DPV \[page 535\]](#)

[DPV security \[page 536\]](#)

[Required information in the job setup \[page 543\]](#)

[Enabling DPV \[page 544\]](#)

[DPV output fields \[page 545\]](#)

[Non-certified mode \[page 549\]](#)

[DPV performance \[page 550\]](#)

[DPV information in US Addressing Report \[page 552\]](#)

[DPV No Stats indicators \[page 553\]](#)

[DPV Vacant indicators \[page 554\]](#)

[DPV PO Box Throwback indicator \[page 555\]](#)

[DPV Door Not Accessible indicator \[page 556\]](#)

18.6.6.1.4 Required information in the job setup

Complete the USPS-required components in the transform configuration.

The following list contains the USPS-required options located in the USPS License Information group:

- *Customer Company Name*
- *Customer Company Address*
- *Customer Company Locality*
- *Customer Company Region*
- *Customer Company Postcode1*
- *Customer Company Postcode2*

Parent topic: [USPS DPV \[page 534\]](#)

Related Information

[Benefits of DPV \[page 535\]](#)
[DPV security \[page 536\]](#)
[DPV monthly directories \[page 542\]](#)
[Enabling DPV \[page 544\]](#)
[DPV output fields \[page 545\]](#)
[Non-certified mode \[page 549\]](#)
[DPV performance \[page 550\]](#)
[DPV information in US Addressing Report \[page 552\]](#)
[DPV No Stats indicators \[page 553\]](#)
[DPV Vacant indicators \[page 554\]](#)
[DPV PO Box Throwback indicator \[page 555\]](#)
[DPV Door Not Accessible indicator \[page 556\]](#)

18.6.6.1.5 Enabling DPV

i Note

DPV is required for CASS.

In addition to the required customer company information that you enter into the USPS License Information group, set the following options to perform DPV processing:

1. Open the USA Regulatory Address Cleanse transform.
2. Open the [Options](#) tab. Expand the Assignment Options group, and select Yes for the [Enable DPV](#) option.
3. In the Reference Files group, enter the path for your DPV directories in the [DPV Path](#) option.

i Note

DPV can run only when the location for all the DPV directories have been entered and none of the DPV directory files have been renamed.

4. Set a directory for the DPV log file in the [USPS Path](#) option. Use the substitution variable `$CertificationLogPath` if you have it set up.
5. In the Report and Analysis group, select Yes for the [Generate Report Data](#) option.

Task overview: [USPS DPV \[page 534\]](#)

Related Information

[Benefits of DPV \[page 535\]](#)
[DPV security \[page 536\]](#)

[DPV monthly directories \[page 542\]](#)
[Required information in the job setup \[page 543\]](#)
[DPV output fields \[page 545\]](#)
[Non-certified mode \[page 549\]](#)
[DPV performance \[page 550\]](#)
[DPV information in US Addressing Report \[page 552\]](#)
[DPV No Stats indicators \[page 553\]](#)
[DPV Vacant indicators \[page 554\]](#)
[DPV PO Box Throwback indicator \[page 555\]](#)
[DPV Door Not Accessible indicator \[page 556\]](#)

18.6.6.1.6 DPV output fields

To obtain DPV output information from your job processing, include specific output fields in your output file setup.

The following table contains the output fields available for reporting DPV results.

Field	Description
<i>DPV_CMRA</i>	Indicates that the address is a Commercial Mail Receiving Agency (CMRA). <i>L</i>: Address triggered DPV locking. <i>N</i>: Address is not a CMRA <i>Y</i>: Address is a valid CMRA <i>blank</i>: <i>Enable DPV</i> is set to <i>NO</i>, DPV processing is currently locked, or transform cannot assign input address.
<i>DPV_Door_Not_Accessible</i>	Indicates that the address is a designated Door Not Accessible (DNA) address. <i>Y</i>: Address is a DNA address. <i>N</i>: Address is not a DNA address. <i>blank</i>: <i>Enable DPV</i> is set to <i>NO</i>, <code>dpv_door_na.dir</code> is not present, or there is an assignment error. The USPS uses DNA indicators to mark addresses where the carrier cannot access the mail receptacle or door.

Field	Description
<i>DPV_Footnote</i>	Indicates the DPV information about the record. i Note CASS requires DPV footnotes. • <i>AA</i>: Input address matches to the postcode2 file. • <i>A1</i>: Input address does not match to the postcode2 file. • <i>BB</i>: All input address field values match to DPV. • <i>CC</i>: Input address primary number matches to DPV, but the secondary number does not match (the secondary is present but invalid). • <i>F1</i>: Input address matches a military address. • <i>G1</i>: Input address matches a general delivery address. • <i>M1</i>: Input address primary number is missing. • <i>M3</i>: Input address primary number is invalid. • <i>N1</i>: Input address primary number matches to DPV but the address is missing the secondary number. • <i>P1</i>: Input address is missing the RR or HC Box number. • <i>P3</i>: Input address has an invalid PO, RR, or HC number. • <i>RR</i>: Input address matches to CMRA. • <i>R1</i>: Input address matches to CMRA, but the secondary number is not present. • <i>U1</i>: Input address matches a unique address.
<i>DPV_Non_Delivery_Days</i>	Indicates that there are days when there is no mail delivery. Values include the following: • <i>Y</i>: There are days when there is no mail delivery. • <i>N</i>: There are no days when there is no mail delivery. • <i>blank</i>: <i>Enable DPV</i> is set to <i>NO</i>, The non delivery days directories are not present, or there is an assignment error. The directories for non-delivery days are: • <code>dpv_non_delivery_days_flag.dir</code> • <code>dpv_non_delivery_days.dir</code>

Field	Description
<i>DPV_Non_Delivery_Day_Sun</i>	Indicates a specific day of the week is a non-delivery day.
<i>DPV_Non_Delivery_Day_Mon</i>	Valid values:
<i>DPV_Non_Delivery_Day_Tue</i>	• <i>Y</i>: Indicates a non-delivery day. • <i>N</i>: Indicates that mail is delivered on that particular day but not on all days. • <i>Blank</i>: <i>Enable DPV</i> is set to <i>NO</i>, both of the directory files for non-delivery days are missing, or there was an assignment error.
<i>DPV_Non_Delivery_Day_Wed</i>	Directories for non-delivery days are:
<i>DPV_Non_Delivery_Day_Thu</i>	<code>dpv_non_delivery_days_flag.dir</code> and
<i>DPV_Non_Delivery_Day_Fri</i>	<code>dpv_non_delivery_days.dir</code> .
<i>DPV_No_Secure_Location</i>	Indicates that the address has a No Secure Location (NSL) indicator. • <i>Y</i>: Address is a designated NSL. • <i>N</i>: Address is not a designated NSL. • <i>blank</i>: <i>Enable DPV</i> is set to <i>NO</i>, The NSL directory is not present, or there is an assignment error. The USPS uses NSL indicators to mark addresses where there is no security. For example: • There is no one at the address to collect the mail. • The storefront does not have a mail receptacle. • The business is closed on certain days. The directory for NSL is <code>dpv_no_secure_loc.dir</code>.
<i>DPV_NoStats</i>	Indicates that the address has a No Stats indicator. i Note No Stats means that the address is a vacant property, it receives mail as a part of a drop, or it does not have an established delivery yet. The software outputs DPV No Stats counts to the DPV Summary section of the US Addressing report. • <i>Y</i>: Address flagged as No Stats in DPV data. • <i>N</i>: Address not flagged as No Stats. • <i>blank</i>: <i>Enable DPV</i> is set to <i>NO</i>, The NoStats directory is not present, or there is an assignment error.

Field	Description
<i>DPV_Status</i>	Indicates the DPV status component generated for this record. • <i>D</i>: Primary range is a confirmed delivery point, but secondary range was not available on input. • <i>L</i>: Address triggered DPV locking. • <i>N</i>: Address is not a valid delivery point. • <i>S</i>: Dropped secondary address information or dropped trailing alpha from primary address to match primary range to a confirmed delivery point. • <i>Y</i>: Address is a confirmed delivery point. The primary range and secondary range (if present) are valid. • <i>blank</i>: Either <i>Enable DPV</i> is set to <i>No</i>, DPV processing is currently locked, or transform cannot assign input address.
<i>DPV_Throwback_Indicator</i>	Indicates that the address is a DPV PO Box Throwback. • <i>Y</i>: The address is a DPV PO Box Throwback. • <i>N</i>: The address is not a DPV Throwback. • <i>blank</i>: <i>Enable DPV</i> is set to <i>NO</i>, <code>dpv_po_box_throwback.dir</code> is not present, or there is an assignment error. The indicator identifies street addresses where the delivery point is a street address but mail delivery is to the customer PO Box address.
<i>DPV_Vacant</i>	Indicates that the address is vacant. • <i>Y</i>: Address is vacant. • <i>N</i>: Address is not vacant. • <i>blank</i>: <i>Enable DPV</i> is set to <i>NO</i>, the DPV Vacant directory is not present, or there is an assignment error. The US Addressing report contains DPV Vacant counts in the DPV Summary section.

Parent topic: [USPS DPV \[page 534\]](#)

Related Information

[Benefits of DPV \[page 535\]](#)

[DPV security \[page 536\]](#)

[DPV monthly directories \[page 542\]](#)

[Required information in the job setup \[page 543\]](#)

[Enabling DPV \[page 544\]](#)

[Non-certified mode \[page 549\]](#)

[DPV performance \[page 550\]](#)
[DPV information in US Addressing Report \[page 552\]](#)
[DPV No Stats indicators \[page 553\]](#)
[DPV Vacant indicators \[page 554\]](#)
[DPV PO Box Throwback indicator \[page 555\]](#)
[DPV Door Not Accessible indicator \[page 556\]](#)
[USA Regulatory Address Cleanse output fields](#)

18.6.6.1.7 Non-certified mode

Assign Postcode2 data to addresses that do not DPV confirm.

When addresses do not DPV confirm, you can still assign a Postcode 2 to your data by running in non-certified mode. The mailing does not qualify for CASS, but the addresses are more complete.

To add Postcode2 information to addresses that do not DPV confirm, ensure that you make the following transform settings:

- Disable DPV
- Enable the non-certified option *Assign Postcode2 Not DPV Validated*.

Caution

When you disable DPV processing, the software does not generate the CASS-required documentation and the mailing is not eligible for postal discounts.

[Enabling non-certified mode \[page 550\]](#)

Parent topic: [USPS DPV \[page 534\]](#)

Related Information

[Benefits of DPV \[page 535\]](#)
[DPV security \[page 536\]](#)
[DPV monthly directories \[page 542\]](#)
[Required information in the job setup \[page 543\]](#)
[Enabling DPV \[page 544\]](#)
[DPV output fields \[page 545\]](#)
[DPV performance \[page 550\]](#)
[DPV information in US Addressing Report \[page 552\]](#)
[DPV No Stats indicators \[page 553\]](#)
[DPV Vacant indicators \[page 554\]](#)
[DPV PO Box Throwback indicator \[page 555\]](#)
[DPV Door Not Accessible indicator \[page 556\]](#)

18.6.6.1.7.1 Enabling non-certified mode

To run your job in non certified mode, follow these setup steps:

1. In the Assignment Options group, set the [Enable DPV](#) option to No.
2. In the Non Certified options group, set the [Disable Certification](#) to Yes.
3. In the Non Certified options group, set the [Assign Postcode2 Not DPV Validated](#) to Yes.

⚠ Caution

The software blanks out all Postcode2 information in your data if you disable DPV processing and you disable the [Assign Postcode2 Not DPV Validated](#) option. This includes Postcode2 information provided in your input file.

Task overview: [Non-certified mode \[page 549\]](#)

18.6.6.1.8 DPV performance

The additional time required to perform DPV processing may increase processing time.

Processing time varies with the DPV feature based on operating system, system configuration, and other variables that may be unique to your operating environment.

Decrease the time required for DPV processing by loading DPV directories into system memory before processing. Additionally, manage memory usage by caching the DPV directories.

[Memory usage \[page 551\]](#)

DPV processing may require you to add more memory on your operating system.

[Caching DPV directories \[page 551\]](#)

[Run multiple jobs with DPV \[page 552\]](#)

Add time for a pause between jobs to allow time for the software to release memory.

Parent topic: [USPS DPV \[page 534\]](#)

Related Information

[Benefits of DPV \[page 535\]](#)

[DPV security \[page 536\]](#)

[DPV monthly directories \[page 542\]](#)

[Required information in the job setup \[page 543\]](#)

[Enabling DPV \[page 544\]](#)

[DPV output fields \[page 545\]](#)

[Non-certified mode \[page 549\]](#)

[DPV information in US Addressing Report \[page 552\]](#)

[DPV No Stats indicators \[page 553\]](#)

[DPV Vacant indicators \[page 554\]](#)

[DPV PO Box Throwback indicator \[page 555\]](#)

[DPV Door Not Accessible indicator \[page 556\]](#)

18.6.6.1.8.1 Memory usage

DPV processing may require you to add more memory on your operating system.

SAP recommends a minimum of 768 MB to process with DPV enabled.

To determine the amount of memory required to run with DPV enabled, check the size of the DPV directories (recently about 1 GB¹) and add that to the amount of memory required to run the software. The size of the DPV directories vary depending on the amount of new data in each directory release.

Make sure that your computer has enough memory available before performing DPV processing.

To find the amount of disk space required to cache the directories, see the [Product Availability Matrix \(PAM\)](#) on the SAP Support Portal.

In the PAM, find information about disk space for directories in the appendix “Disk Space Requirements for Referential Data & Cleansing Packages”.

Parent topic: [DPV performance \[page 550\]](#)

Related Information

[Caching DPV directories \[page 551\]](#)

[Run multiple jobs with DPV \[page 552\]](#)

18.6.6.1.8.2 Caching DPV directories

To better manage memory usage when you have enabled DPV processing, choose to cache the DPV directories.

To set up your job for DPV caching, follow these steps:

1. In the Transform Performance group, set the [Cache DPV Directories](#) option to Yes.
2. In the same group, set the [Insufficient Cache Memory Action](#) to one of the following:

¹ The directory size is subject to change each time you install new DPV directories.

Option	Description
Error	Software issues an error and terminates the transform.
Continue	Software attempts to continue initialization without caching.

Task overview: [DPV performance \[page 550\]](#)

Related Information

[Memory usage \[page 551\]](#)

[Run multiple jobs with DPV \[page 552\]](#)

18.6.6.1.8.3 Run multiple jobs with DPV

Add time for a pause between jobs to allow time for the software to release memory.

Add a 10-second pause between job executions when you run multiple DPV jobs and load directories into memory. The 10-second pause allows time for the system to release memory. For more information about setting pauses between jobs properly, see your operating system manual.

If you don't add a 10-second pause between jobs, there may not be enough time for your system to release the memory used for caching the directories from the first job. If there is not enough memory to cache directories, the next job waiting to process may produce an error or access the directories from disk, resulting in performance degradation.

Parent topic: [DPV performance \[page 550\]](#)

Related Information

[Memory usage \[page 551\]](#)

[Caching DPV directories \[page 551\]](#)

18.6.6.1.9 DPV information in US Addressing Report

The US Addressing Report automatically generates when you have enabled reporting in your job. The following sections of the US Addressing Report contain DPV information:

- DPV Return Codes
- Delivery Point Validation (DPV) Summary

For information about the US Addressing Report, or other Data Quality reports, see the *Management Console Guide*.

Parent topic: [USPS DPV \[page 534\]](#)

Related Information

[Benefits of DPV \[page 535\]](#)
[DPV security \[page 536\]](#)
[DPV monthly directories \[page 542\]](#)
[Required information in the job setup \[page 543\]](#)
[Enabling DPV \[page 544\]](#)
[DPV output fields \[page 545\]](#)
[Non-certified mode \[page 549\]](#)
[DPV performance \[page 550\]](#)
[DPV No Stats indicators \[page 553\]](#)
[DPV Vacant indicators \[page 554\]](#)
[DPV PO Box Throwback indicator \[page 555\]](#)
[DPV Door Not Accessible indicator \[page 556\]](#)
[Management Console: Data Quality reports, US Addressing Report](#)

18.6.6.1.10 DPV No Stats indicators

The USPS designates addresses as No Stats when the address is a vacant property, it receives mail as part of a drop, or it doesn't have an established delivery yet.

SAP Data Services uses the No Stats table when you have DPV or DSF2 enabled in the transform. Data Services includes the No Stats table with the DPV directories.

When you enable DPV, Data Services automatically checks for the No Stats table in the directory folder that you indicate in the transform. If the No Stats table is missing, Data Services skips No Stats processing and doesn't issue an error message.

To obtain No Stats information from Data Services, enable DPV and include the [DPV_NoStats](#) output field in your output field setup. Data Services populates the No Stats output field as follows:

- **Y**: Address is flagged with No Stats.
- **N**: Address isn't flagged with No Stats.
- **blank**: DPV isn't enabled, the No Stats table is missing, or there was an address correction error.

Parent topic: [USPS DPV \[page 534\]](#)

Related Information

[Benefits of DPV \[page 535\]](#)
[DPV security \[page 536\]](#)
[DPV monthly directories \[page 542\]](#)
[Required information in the job setup \[page 543\]](#)
[Enabling DPV \[page 544\]](#)
[DPV output fields \[page 545\]](#)
[Non-certified mode \[page 549\]](#)
[DPV performance \[page 550\]](#)
[DPV information in US Addressing Report \[page 552\]](#)
[DPV Vacant indicators \[page 554\]](#)
[DPV PO Box Throwback indicator \[page 555\]](#)
[DPV Door Not Accessible indicator \[page 556\]](#)

18.6.6.1.11 DPV Vacant indicators

The USPS assigns a DPV Vacant indicator to addresses that had an active DPV but now it is vacant (for over 90 days), it doesn't receive mail delivery, and it can receive mail delivery in the future.

Vacancy doesn't apply to seasonal addresses.

Obtain vacant indicators from SAP Data Services by enabling DPV and DSF2, and including the output field *DPV_Vacant*.

The software includes DPV vacant counts and percentage in the DPV Summary section of the US Addressing report.

Parent topic: [USPS DPV \[page 534\]](#)

Related Information

[Benefits of DPV \[page 535\]](#)
[DPV security \[page 536\]](#)
[DPV monthly directories \[page 542\]](#)
[Required information in the job setup \[page 543\]](#)
[Enabling DPV \[page 544\]](#)
[DPV output fields \[page 545\]](#)
[Non-certified mode \[page 549\]](#)
[DPV performance \[page 550\]](#)
[DPV information in US Addressing Report \[page 552\]](#)
[DPV No Stats indicators \[page 553\]](#)

[DPV PO Box Throwback indicator \[page 555\]](#)

[DPV Door Not Accessible indicator \[page 556\]](#)

18.6.6.1.12 DPV PO Box Throwback indicator

The USPS uses the PO Box Throwback table to identify delivery points that are street addresses but mail delivery is to a PO Box address instead.

The USPS introduced the PO Box Throwback indicator for the DPV and DSF2 programs to help grow the package services business. PO Box Throwback indicators help carriers make sure that mail reaches the intended destinations.

SAP Data Services uses the PO Box Throwback table named `dpv_po_box_throwback.dir`, which is included with the DPV directories.

The `dpv_po_box_throwback.dir` file is optional. Therefore, if you don't download the table with your DPV files, Data Services doesn't issue an error.

To enable PO Box Throwback indication in Data Services, include the output field *DPV_Throwback_Indicator* in your output field configuration. Data Services populates the field as follows:

- *Y*: Address is flagged as PO Box Throwback.
- *N*: Address isn't flagged as PO Box Throwback.
- *blank*: DPV isn't enabled, you don't have the PO Box Throwback table, or there's an error in address correction.

Currently, the PO Box Throwback table is also a part of DSF2 processing in the DSF2 augment directories. The PO Box Throwback table contains the same content in the DPV and DSF2 directories. Because DPV is required for DSF2, Data Services uses the same table for both processes. However, if you enable DPV but disable DSF2, Data Services doesn't output an indicator to the DSF2 output field `DSF2_Throwback_Indicator`.

The US Addressing Report contains a row in the DPV Summary section for DPV throwback information. The row named **Throwback Addresses** contains the record count and percentage of PO Box Throwback addresses in your database.

Parent topic: [USPS DPV \[page 534\]](#)

Related Information

[Benefits of DPV \[page 535\]](#)

[DPV security \[page 536\]](#)

[DPV monthly directories \[page 542\]](#)

[Required information in the job setup \[page 543\]](#)

[Enabling DPV \[page 544\]](#)

[DPV output fields \[page 545\]](#)

[Non-certified mode \[page 549\]](#)

[DPV performance \[page 550\]](#)

[DPV information in US Addressing Report \[page 552\]](#)

[DPV No Stats indicators \[page 553\]](#)

[DPV Vacant indicators \[page 554\]](#)

[DPV Door Not Accessible indicator \[page 556\]](#)

18.6.6.1.13 DPV Door Not Accessible indicator

The USPS uses Door Not Accessible (DNA) indicators to mark addresses where the carrier can't knock on the door for mail delivery.

For example, DNA indicates addresses with the following characteristics:

- Carrier can't physically access a residence or building because it is on a rural route or highway contract route with a long driveway.
- Address is in a gated community that the carrier can't access.

SAP Data Services uses the DNA table named `dpv_door_na.dir`, which is included with your DPV directories.

To enable DNA in Data Services, enable DPV in your job setup, and include the `dpv_door_na.dir` with your DPV directories. With DPV enabled, Data Services automatically checks for the DNA table in the directory folder that you indicate in the transform. If you don't install the table, Data Services skips DNA processing and doesn't issue an error message.

To obtain DNA information from Data Services, include the output field *DPV_Door_Not_Accessible* in your output field configuration. Data Services populates the field as follows:

- *Y*: Address is flagged as DNA.
- *N*: Address isn't flagged as DNA.
- *blank*: DPV isn't enabled, you don't have the DNA table, or there's an error in address correction.

Additionally, the US Addressing Report contains a row in the DPV Summary section. The row named **DPV Door Not Accessible** contains the record count and percentage of DNA addresses in your database.

Parent topic: [USPS DPV \[page 534\]](#)

Related Information

[Benefits of DPV \[page 535\]](#)

[DPV security \[page 536\]](#)

[DPV monthly directories \[page 542\]](#)

[Required information in the job setup \[page 543\]](#)

[Enabling DPV \[page 544\]](#)

[DPV output fields \[page 545\]](#)

[Non-certified mode \[page 549\]](#)

[DPV performance \[page 550\]](#)

[DPV information in US Addressing Report \[page 552\]](#)

[DPV No Stats indicators \[page 553\]](#)

[DPV Vacant indicators \[page 554\]](#)

[DPV PO Box Throwback indicator \[page 555\]](#)

18.6.6.2 LACSLink

LACSLink updates addresses when the physical address does not move but the address is changed.

An address change can happen with municipality address requirements, street renaming, or PO Box renumbering. For example, to make it easier for police, fire, ambulance, and postal personnel to locate rural addresses, a municipality requires that all rural addresses be assigned a street name.

LACSLink technology ensures that the data remains private and secure, and at the same time gives you easy access to the data. LACSLink is an integrated part of address processing; it is not an extra step. You cannot obtain new addresses unless you have the old address data.

LACSLink is a USPS product that is available for U.S. records with the USA Regulatory Address Cleanse transform only. LACSLink processing is required for CASS certification.

Parent topic: [Beyond the basic address cleansing \[page 533\]](#)

Related Information

[USPS DPV \[page 534\]](#)

[SuiteLink \[page 567\]](#)

[USPS DSF2* \[page 570\]](#)

[NCOALink and Move Update \[page 581\]](#)

[USPS eLOT \[page 598\]](#)

[Early Warning System \(EWS\) \[page 599\]](#)

[USPS RDI \[page 601\]](#)

[GeoCensus \(USA Regulatory Address Cleanse \) \[page 604\]](#)

[Z4Change \[page 608\]](#)

[Suggestion lists \[page 610\]](#)

[Multiple data source statistics reporting \[page 613\]](#)

[How LACSLink works \[page 558\]](#)

[Control memory usage for LACSLink processing \[page 566\]](#)

[To disable LACSLink \[page 563\]](#)

[LACSLink security \[page 558\]](#)

18.6.6.2.1 How LACSLink works

When the software processes data with LACSLink enabled, it follows a specific process.

LACSLink provides a new address when one is available. The software performs the following steps when you enable LACSLink:

1. The USA Regulatory Address Cleanse transform standardizes the input address.
2. The transform looks for a matching address in the LACSLink data.
3. If a match is found, the transform outputs the LACSLink-converted address and other LACSLink information.

Related Information

[Control memory usage for LACSLink processing \[page 566\]](#)

[LACSLink \[page 557\]](#)

18.6.6.2.2 LACSLink security

To ensure that the LACSLink data is not misused, the USPS has processes that monitor the use of LACSLink.

When a company purchases the LACSLink functionality, the USPS requires them to sign a legal agreement. The agreement states that the company agrees not to misuse the LACSLink product. If the company abuses the LACSLink product, the USPS can prohibit the user from using LACSLink in the future.

18.6.6.2.2.1 LACSLink false positive addresses

The USPS includes false positive addresses in the LACSLink directories as an added security to prevent LACSLink abuse.

A false positive address is a fake address that the USPS places in the LACSLink directories. When the software encounters a false positive address in your data, it can stop LACSLink processing or continue processing based on your LACSLink user type. Your action when the software encounters a false positive also depends on your user type.

The following table explains the behaviors for each user type.

User type	Software behavior	Action
End users without a Stop Processing Alternative agreement	LACSLink directories lock and LACSLink processing terminates. Job continues to run without LACSLink.	1. Unlock LACSLink processing by obtaining a LACSLink unlock code from SAP Support. 2. Send false positive logs to the USPS to release the mailing list.

User type	Software behavior	Action
End user with Stop Processing Alternative agreement	LACSLink processing continues.	Send false positive logs to the USPS to release the mailing list
Service providers	LACSLink processing continues.	Send false positive logs to the USPS to release the mailing list.

Related Information

[Stop Processing Alternative \[page 538\]](#)

[Obtaining LACSLink unlock code from SAP \[page 560\]](#)

[Sending LACSLink false positive logs to the USPS \[page 561\]](#)

18.6.6.2.2.2 LACSLink false positive logs

The software generates a false-positive log file any time it encounters a LACSLink false positive record.

The software creates a separate log file for each mailing list that contains a false positive record. If multiple false positives exist within one mailing list, the software writes them all to the same log file.

The software stores LACSLink log files in the directory specified for the [USPS Log Path](#) in the Reference Files group.

i Note

Ensure that the directory that you enter is writable. The software issues an error if it cannot write to the specified path.

The software automatically names LACSLink false positive logs `lacs1###.log`, where `###` is a number between 001 and 999. For example, the first log file generated is `lacs1001.log`, the next one is `lacs1002.log`, and so on.

i Note

When you set the degree of parallelism (DOP) to greater than 1, the software generates 1 log per thread that contains a false positive.

Related Information

[Performance Optimization Guide: Using parallel execution](#)

18.6.6.2.2.3 LACSLink locking for end users

If you are an end user without a Stop Processing Agreement with the USPS and the software encounters a false positive record in your database, it stops LACSLink processing for the remainder of the data flow.

The following locking behavior is applicable for end users or users who are DSF2 licensees that have DSF2 disabled in the job.

In addition to terminating LACSLink processing, the software also takes the following actions:

- Marks the record as a false positive address.
- Issues a message in the error log that a LACSLink false positive address was encountered.
- Includes the false positive address and lock code in the error log.
- Continues processing your data flow without LACSLink processing.
- Generates a lock code.
- Generates a false positive log.
- Generates a US Regulatory Locking Report that contains the false positive address and the lock code. (Report generation must be enabled in the USA Regulatory Address Cleanse transform.)

To restore LACSLink functionality, obtain a LACSLink unlock code from SAP Support following the steps in [Obtaining LACSLink unlock code from SAP \[page 560\]](#).

18.6.6.2.2.4 Obtaining LACSLink unlock code from SAP

When the transform locks LACSLink processing because it has found a false positive address in your data, you unlock LACSLink processing by obtaining an unlock code from SAP.

The following steps apply to end users who do not have a Stop Processing Alternative agreement with the USPS.

Before you perform the following steps, write down the lock code and the locking address from the `lacsx.txt` file. The file is located in the directory you specify in the [LACSLink Path](#) option in the Reference Files settings of the transform.

1. Log into the SAP Support Portal and go to <https://support.sap.com/en/my-support/keys.html>.
2. Select the **USPS Unlock Code link** at the top of the page.
3. Select **Get your USPS Unlock Codes**.
4. Click [Search](#) and select an applicable Data Services system from the list.
5. Enter the lock code that you wrote down from the `lacsx.txt` file.
6. Select LACSLink as the lock type.
7. Select BOJ-EIM-DS as the component.
8. Enter the address that caused the lock that you wrote down from the `lacsx.txt` file.
9. Attach the `lacs1####.log` file.
10. Click [Submit](#).

The utility displays the unlock code.

i Note

If the unlock code could not be generated, the system sends a message to a Technical Customer Assurance engineer who generates an unlock code for you during regular business hours.

11. Open the `lacsw.txt` file, delete all content, and paste the unlock code into the file.

i Note

The `lacsw.txt` file is different than the `lacsx.txt` file.

12. Remove the record that caused the lock from the applicable database.
13. Delete the `lacs1####.log` file before processing the list again.

i Note

Before you can process the list for production, obtain a list release from the USPS following the process in [Sending LACSLink false positive logs to the USPS \[page 561\]](#).

You can use the unlock code only one time. If the software detects another false-positive, even if it is the same record, you have to retrieve a new LACSLink unlock code. Therefore, before you process the list again, make sure that you perform steps 11 and 12.

Send the false positive log to the USPS to obtain release of the mailing list.

18.6.6.2.2.5 Sending LACSLink false positive logs to the USPS

If you receive a message from the software that LACSLink false positive addresses are present in your address list, obtain a list release from the USPS before you use the list again.

This process applies to all LACSLink user types, regardless of whether they have a Stop Processing Alternative agreement with the USPS.

1. Send an email to the USPS at `dsf2stop@usps.gov`. Include the following information:
 - Type "LACSLink False Positive" as the subject line
 - Attach the `lacs1####.log` file or files.

The USPS NCSC uses the information to determine whether it can release the list to the mailer.

2. After the USPS NCSC has released the list that contained the locked or false positive record:
 - Delete the corresponding log file or files
 - Delete the false positive record from your database
3. Reprocess the database.

If you are an end user who does not have a Stop Processing Alternative agreement, follow the steps to obtain the LACSLink unlock code from SAP Support.

Related Information

[Obtaining LACSLink unlock code from SAP \[page 560\]](#)

18.6.6.2.3 Conditions for address processing

To save processing time, the software performs LACSLink processing only under certain conditions.

The transform does not process all of your addresses with LACSLink when you have it enabled. The following list contains conditions under which the software passes your data to LACSLink processing:

- The software finds that the USPS flagged the address in the address directories as a LACS-convertible record.
- The software finds that the USPS has not flagged the address for LACSLink, but the software assigns the address for a rural route or highway contract by default
- The software does not find the address in the address directory, but the record contains enough information to qualify it for LACSLink processing.

❖ Example

The following table shows an address that is in the address directory as a LACS-convertible address.

Original address	After LACSLink conversion
RR2 BOX 204	463 SHOWERS RD
DU BOIS PA 15801	DU BOIS PA 15801-66675

18.6.6.2.4 Sample transform configuration

LACSLink processing is enabled by default in the sample transform configuration because it is required for CASS certification. The sample transform configuration is named USARegulatory_AddressCleanse and is found under the USA_Regulatory_Address_Cleanse group in the Object Library.

18.6.6.2.5 LACSLink directory files

SAP ships the LACSLink directory files with the U.S. National Directory update.

The LACSLink directory files require about 600 MB² of additional hard drive space. The LACSLink directories include the following files:

² The directory size changes over time. See the PAM (Product Availability Matrix) for the most recent directory size.

- `lacsx.txt`
- `lacsx.txt`
- `lacsx.ll`
- `lacsx.ll`

⚠ Caution

The LACSLink directories must reside on the hard drive in the same directory as the LACSLink supporting files. Do not rename any of the files. LACSLink will not run if the file names are changed.

LACSLink directories expire in 105 days. LACSLink directories must have the same date as the other directories that you are using from the U.S. National Directories.

18.6.6.2.6 Enabling LACSLink

LACSLink is enabled by default in the USA Regulatory Address Cleanse transform. If you need to re-enable the option, follow these steps:

1. Open the USA Regulatory Address Cleanse transform and open the [Options](#) tab.
2. Expand the Processing Options group
3. select Yes in the [Enable LACSLink](#) option.
4. Enter the LACSLink path for the [LACSLink Path](#) option In the Reference Files group. You can use the substitution variable `$$RefFilesAddressCleanse` if you have it set up.
5. Complete the required fields in the USPS License Information group.

18.6.6.2.6.1 Required information in the job setup

Because CASS certification requires LACSLink, enter the required USPS license information when you set up the USA Regulatory Address Cleanse transform.

The following list contains the USPS-required options located in the USPS License Information group:

- [Customer Company Name](#)
- [Customer Company Address](#)
- [Customer Company Locality](#)
- [Customer Company Region](#)
- [Customer Company Postcode1](#)
- [Customer Company Postcode2](#)
- [Customer Company Phone](#)

18.6.6.2.6.2 To disable LACSLink

LACSLink is enabled by default in the USA Regulatory Address Cleanse transform configuration because it is required for CASS processing. Therefore, you must disable CASS certification in order to disable LACSLink.

1. In the USA Regulatory Address Cleanse transform configuration, open the [Options](#) tab.
2. Open the Non Certified Options group.
3. Select Yes for the [Disable Certification](#) option.
4. Open the Assignment Option group.
5. Select No for the [Enable LACSLink](#) option.

Related Information

[LACSLink \[page 557\]](#)

18.6.6.2.6.3 Reasons for errors

If you encounter errors during LACSLink job processing, it may be because of your job setup.

If your job setup is missing information in the USPS License Information group, and you have DPV and/or LACSLink enabled in your job, the transform issues error messages based on the following situations:

Reason for error	Description
Missing required options	When your job setup does not include the required parameters in the USPS License Information group, and you have DPV and/or LACSLink enabled in your job, the software issues a verification error.
Unwritable Log File directory	If the path that you specified for the USPS Log Path option in the Reference Files group is not writable, the software issues an error.

18.6.6.2.7 LACSLink output fields

Several output fields are available for reporting LACSLink processing results.

Ensure that you enable LACSLink and include the following output fields in your job setup so that the transform posts LACSLink information to output.

LACSLink output fields and descriptions

Field name	Length	Description
LACSLink_Query	50	Returns the pre-conversion address, populated only when LACSLink is enabled and a LACSLink lookup was attempted. Returns <i>blank</i> when it did not attempt a LACSLink lookup. The pre-conversion address is in the standard USPS format (as shown in USPS Publication 28). However, when an address has both a unit designator and secondary unit, the transform replaces the unit designator by the pound "#" character.
LACSLink_Return_Code	2	Returns the match status for LACSLink processing: • A: LACSLink record match. The transform provided a converted address in the address data fields. • 00: No match and no converted address. • 09: LACSLink matched an input address to an old address, which is a "high-rise default" address. The transform did not provide a new address. • 14: Found a LACSLink record, but couldn't convert the data to a deliverable address. • 92: LACSLink record matched after dropping the secondary number from input address. • <i>blank</i>: No LACSLink lookup attempted.

Field name	Length	Description
LACSLink_Indicator	1	Returns the conversion status of addresses processed by LACSLink. • Y: Address converted by LACSLink. The LACSLink_Return_Code value is A. • N: Address looked up with LACSLink but not converted. • F: Address is a false-positive. • S: LACSLink conversion made, but the transform dropped the secondary information. • <i>blank</i>: No LACSLink lookup attempted.

18.6.6.2.8 Control memory usage for LACSLink processing

The transform performance improves when you cache the LACSLink directories.

Set the software to cache LACSLink directories in the Transform Performance section of the transform configuration.

To find the amount of disk space required to cache the directories, see the [Product Availability Matrix \(PAM\)](#) on the SAP Support Portal.

Find information about disk space for directories in the appendix “Disk Space Requirements for Referential Data & Cleansing Packages.”

The software issues a verification error message at run-time and the transform terminates if:

- You do not have adequate system memory to load the LACSLink directories
- You set the [Insufficient Cache Memory Action](#) option to [Error](#)

If you select to continue, the transform attempts to continue LACSLink processing without caching.

Open the [Options](#) tab of your USA Regulatory Address Cleanse transform configuration in your data flow. Follow these steps to load the LACSLink directories into your system memory:

1. Open the Transform Performance option group.
2. Select [Yes](#) for the [Cache LACSLink Directories](#) option.

Related Information

[LACSLink \[page 557\]](#)

18.6.6.2.9 LACSLink information in US Addressing Report

The US Addressing Report contains sections with LACSLink information.

The US Addressing Report automatically generates when you enable reporting in your job. The following table lists the LACSLink sections in the US Addressing Report.

Section	Information
Locatable Address Conversion (LACSLink) Summary	Contains record counts and percentages for the following information: • LACSLink converted addresses • Addresses not LACSLink converted
LACSLink Return Codes	Contains record counts and percentages for the following LACSLink return code types: • Converted • Secondary dropped • No match • Can't convert • High-rise default

18.6.6.2.10 USPS Form 3553

The USPS Form 3553 contains LACSLink counts.

If you have LACSLink enabled in the job, the LACS/LACSLink field in the USPS Form 3553 shows the number of records that have a LACSLink Indicator of **Y** or **S**. If you do not enable LACSLink processing, the LACS/LACSLink field shows the count only for the LACS code.

18.6.6.3 SuiteLink

Use the SuiteLink directory information to add suite numbers and other secondary data to high-rise business addresses.

SuiteLink information is in a USPS directory that consists of multiple files of specially indexed address information. For example, the SuiteLink directories contain secondary numbers and unit designators. The SuiteLink data is for locations identified as high-rise default buildings.

When you use the SuiteLink directories, the software adds secondary address information to your addresses, and sorts more mail pieces by delivery sequence. The additional data, and the delivery sequence sortation ensures that the USPS delivers more mail pieces with accuracy and speed.

SuiteLink is required for CASS

SuiteLink is required when you process in CASS mode, and you set the *Disable certification* option to *No*. If you disable SuiteLink in your job setup but you are in CASS mode, the software issues an error message and stops processing.

Parent topic: [Beyond the basic address cleansing \[page 533\]](#)

Related Information

[USPS DPV \[page 534\]](#)

[LACSLink \[page 557\]](#)

[USPS DSF2* \[page 570\]](#)

[NCOALink and Move Update \[page 581\]](#)

[USPS eLOT \[page 598\]](#)

[Early Warning System \(EWS\) \[page 599\]](#)

[USPS RDI \[page 601\]](#)

[GeoCensus \(USA Regulatory Address Cleanse \) \[page 604\]](#)

[Z4Change \[page 608\]](#)

[Suggestion lists \[page 610\]](#)

[Multiple data source statistics reporting \[page 613\]](#)

18.6.6.3.1 Benefits of SuiteLink

The addition of secondary number information to your addresses allows for the most efficient and cost-effective delivery sequencing and postage discounts.

If your business depends on website, mail, or in-store orders from customers, use SuiteLink as a powerful money-saving tool. Also know that your customers who reside in high-rises that house several businesses, receive their marketing materials, bank statements, and orders right at their door.

i Note

SuiteLink is required for CASS.

18.6.6.3.2 How SuiteLink works

The software uses the data in the SuiteLink directories to add suite numbers to an address.

The software matches a company name, a known high-rise address, and the CASS-certified postcode2 in your database to data in SuiteLink. When there is a match, the software creates a complete business address that includes the suite number.

❖ Example

Follow a data record through the SuiteLink process.

1. The software standardizes the address elements based on the standardization options.
2. SuiteLink recognizes the address is a high-rise.
3. SuiteLink uses the high-rise information and the firm name to assign a correct suite number.
4. SuiteLink uses the high-rise information and the firm name to assign a correct ZIP+4 code.

Input record

Telera
910 E Hamilton Ave Fl2
Campbell CA 95008 0610

Output record

TELERA
910 E HAMILTON AVE STE 200
CAMPBELL CA 95008 0625

18.6.6.3.3 SuiteLink directory

When you use the SuiteLink directory, ensure that you know the requirements for use.

Enter the `zip4us.dir` path in the [Address Directory1](#) option of the Reference Files group in the USA Regulatory Address Cleanse transform. For example, use the December 2018 SuiteLink directory only with the December 2018 `zip4us.dir` directory.

You cannot use a SuiteLink directory that is older than 60 days based on its release date. The software warns you 15 days before the directory expires. As with all directories, the software won't process your records with an expired SuiteLink directory.

18.6.6.3.4 Enabling SuiteLink

SuiteLink is enabled by default in any of the sample transform configurations that are set up to be CASS-compliant (and the Disable certification option is set to No). For example, SuiteLink is enabled if you use the USA Regulatory_AddressCleanse sample transform configuration.

1. Open the USA Regulatory Address Cleanse transform in your data flow.
2. Open the [Options](#) tab.
3. Expand the Assignment Options group and set the [Enable SuiteLink](#) option to Yes.

4. In the Reference Files group, enter the SuiteLink directory path in the [SuiteLink Path](#) option. You can use the substitution variable `$$RefFilesAddressCleanse` if you have it set up with the directory location that contains your SuiteLink directories.
5. Optional: In the Transform Performance option group, set the [Cache SuiteLink Directories](#) option to Yes so that the SuiteLink directories are cached in memory.

i Note

Ensure that you have sufficient RAM to cache the SuiteLink directories before you enable this option.

18.6.6.3.5 Improve SuiteLink processing speed

To improve processing speed, load SuiteLink directories into memory.

To load SuiteLink directories into memory, set [Cache SuiteLink Directories](#) in the Transform Performance group to [Yes](#).

18.6.6.3.6 SuiteLink return codes in US Addressing Report

SuiteLink return code information is available in the SuiteLink Return Codes section of the US Addressing Report.

The US Addressing Report shows the record count and percentage for the following return codes:

A = Secondary exists and assignment made

00 = Lookup was attempted but no assignment

18.6.6.4 USPS DSF2®

DSF2 is a USPS-licensed product that you can use to validate addresses, add delivery sequence information, and add DSF2 address attributes to addresses.

Two DSF2 features are supported in Data Services:

- DSF2 Augment in the USA Regulatory Address Cleanse transform
- DSF2 Walk Sequence in the DSF2 Walk Sequencer transform

i Note

USPS DSF2 data is available only to USPS-certified DSF2 licensees.

Parent topic: [Beyond the basic address cleansing \[page 533\]](#)

Related Information

[USPS DPV \[page 534\]](#)
[LACSLink \[page 557\]](#)
[SuiteLink \[page 567\]](#)
[NCOALink and Move Update \[page 581\]](#)
[USPS eLOT \[page 598\]](#)
[Early Warning System \(EWS\) \[page 599\]](#)
[USPS RDI \[page 601\]](#)
[GeoCensus \(USA Regulatory Address Cleanse \) \[page 604\]](#)
[Z4Change \[page 608\]](#)
[Suggestion lists \[page 610\]](#)
[Multiple data source statistics reporting \[page 613\]](#)
[DSF2 Walk sequencing \[page 576\]](#)

18.6.6.4.1 Validating addresses

DSF2 helps reduce the quantity of undeliverable-as-addressed (UAA) mail and keeps mailing costs down. DSF2 uses DPV® to validate addresses and identify inaccurate or incomplete addresses.

Related Information

[DPV overview \[page 534\]](#)

18.6.6.4.2 Adding address attributes

DSF2 adds address attributes (information about the addresses) to your data. Use the attribute information to create more targeted mailings.

18.6.6.4.3 Adding delivery sequence information

DSF2 adds delivery sequence information to your data, which you can use to qualify for walk-sequence discounts. This information is sometimes called walk sequencing or pseudo sequencing.

Related Information

[Walk sequencing \[page 576\]](#)

[Pseudo sequencing \[page 576\]](#)

18.6.6.4.4 Benefits of DSF2

Those who want to target their mail to specific types of addresses and those who want to earn additional postal discounts will appreciate what DSF2 can do.

The DSF2 address-attribute data provides mailers with knowledge about the address above and beyond what is necessary to accurately format the addresses. Address-attribute data allows mailers to produce more targeted mailings.

For example, If you plan to send out a coupon for your lawn-care service business, you do not want to send it to apartment dwellers (they may not have a lawn). You want your coupon to go to residential addresses that are not centralized in an apartment building.

With the DSF2 information you can walk-sequence your mailings to achieve the best possible postal discounts by using the DSF2 Walk Sequencer transform.

18.6.6.4.5 Becoming a DSF2 licensee

Before you can perform DSF2 processing in the software, you must complete the USPS DSF2 certification procedures and become licensed by the USPS.

Part of certification is processing test jobs in Data Services to prove that the software complies with the license agreement. When you are ready to take these tests, contact SAP BusinessObjects Business User Support to obtain access to the DSF2 features in Data Services.

For more information, see the topic “DSF2 Certification” in the *Reference Guide* (in the “USPS Certifications” section) .

18.6.6.4.6 DSF2 directories

DSF2 processing requires the following data:

Data	Notes
DPV directories	The software uses DPV directories to verify addresses and identify inaccurate addresses. SAP supplies the DPV directories with the U.S. National Directory delivery. i Note DPV directories are included with the DSF2 tables. Do not use the DPV directories included with the DSF2 tables. Use the DPV directories from SAP with the U.S. National Directory delivery.
eLOT directories	The software uses eLOT directories to assign walk sequence numbers. SAP supplies the eLOT directories with the U.S. National Directory delivery. i Note eLOT directories are included with the DSF2 tables. Do not use the eLOT directories included with the DSF2 tables. Use the eLOT directories from SAP with the U.S. National Directory delivery.
DSF2 tables	The software uses DSF2 tables to assign address attributes. i Note DSF2 tables are supplied by the USPS and not SAP. In addition, the DSF2 tables include DPV and eLOT directories. Do not use the DPV and eLOT directories included with the DSF2 tables. Use the DPV and eLOT directories from SAP with the U.S. National Directory delivery.
Delivery statistics file	The software uses the delivery statistics file to provide counts of business and residential addresses per ZIP Code (Postcode1) per Carrier Route (Sortcode). SAP supplies the delivery statistics file with the U.S. National Directory delivery.

You must specify the location of these directory files in the USA Regulatory Address Cleanse transform, except for the delivery statistics file. Set the location of the delivery statistics file (dsf.dir) in the DSF2 Walk Sequencer transform. Also, to meet DSF2 requirements, you must install updated directories monthly.

18.6.6.4.7 DSF2 augment processing

Set up DSF2 augment processing in the USA Regulatory Address Cleanse transform.

DSF2 processing requires DPV information, therefore, enable DPV in your job setup.

If you plan to use the output information from the DSF2 augment processing for walk sequence processing, you must also enable eLOT.

i Note

DSF2 augment is available only in batch mode. You cannot add augment information to your data in real time.

18.6.6.4.7.1 DSF2 Augment directory expiration

The DSF2 directories are distributed monthly. You must use the DSF2 directories with U.S. National directories that are labeled for the same month. For example, the May 2011 DSF2 directories can be used with only the May 2011 National directories.

The DSF2 Augment data expires in 60 days instead of the 105 day expiration for the U.S. National directories. Because directories must all have the same base date (MM/YYYY), DSF2 users who have Yes set for the Enable DSF2 Augment option must update all of the U.S. National directories and other directories they use (for example, LACSLink or DPV) at the same time as the DSF2 Augment directories. The software will remind users to update the directories with a warning message that appears 15 days before the directory expires.

→ Remember

As with all directories, the software will not process your records with expired DSF2 directories.

18.6.6.4.7.2 Identifying the DSF2 licensee

When you perform DSF2 processing, you must provide the following information: The DSF2-licensed company and the client for whom the company is processing this job.

You must complete the following options in the USPS License Information group for DSF2 processing:

- DSF2 Licensee ID
- Licensee Name
- List Owner NAICS Code
- List ID
- Customer Company Name
- Customer Company Address
- Customer Company Locality
- Customer Company Region
- Customer Company Postcode1
- Customer Company Postcode2
- List Received Date
- List Return Date

i Note

If you are performing DSF2 and NCOALink processing in the same instance of the USA Regulatory Address Cleanse transform, then the information that you enter in the USPS License Information group must apply to both DSF2 and NCOALink processing. If, for example, the [List ID](#) is different for DSF2 and NCOALink, you will need to include two USA Regulatory Address Cleanse transforms: One for NCOALink and another for DSF2.

18.6.6.4.7.3 Enabling DSF2 Augment

Before you can process with DSF2, you must first become a certified licensee.

In addition to the required customer company information that you enter into the USPS License Information group, set the following options to perform DSF2 Augment processing:

1. In the USA Regulatory Address Cleanse transform, open the [Options](#) tab.
2. Expand the Report and Analysis group and set the [Generate Report Data](#) option to Yes.
3. Expand the Reference Files group and enter the path for the options [DSF2 Augment Path](#), [DPV Path](#), and [eLOT Directory](#), or use the \$\$RefFilesAddressCleanse substitution variable if you have it set up.
4. Also in the Reference Files group, enter a path for the [USPS Log Path](#) option, or use the \$CertificationLogPath substitution variable if you have it set up.
5. Optional. Expand the Transform Performance group and set the [Cache DPV Directories](#) and [Cache DSF2 Augment Directories](#) to Yes.
6. Expand the Assignment Options group and set the [Enable DSF2 Augment](#), [Enable DPV](#), and [Enable eLOT](#) to Yes.
7. Include the DSF2 address attributes output fields in your output file setup.

18.6.6.4.7.4 DSF2 output fields

When you perform DSF2 Augment processing in the software, address attributes are available in the following output fields for every address that was assigned. Be sure to include the fields containing information you'll need in your output file setup:

- DSF2_Business_Indicator
- DSF2_Delivery_Type
- DSF2_Drop_Count
- DSF2_Drop_Indicator
- DSF2_Educational_Ind
- DSF2_LACS_Conversion_Ind
- DSF2_Record_Type
- DSF2_Seasonal_Indicator
- DSF2_Throwback_Indicator

Note

A blank output in any of these fields means that the address was not looked up in the DSF2 directories.

Related Information

[USA Regulatory Address Cleanse input fields](#)

[USA Regulatory Address Cleanse output fields](#)

18.6.6.4.7.5 Improving DSF2 processing speed

You can cache DSF2 data to improve DSF2 processing speed.

To cache DSF2 data, set the *Cache DSF2 Augment Directories* option in the Transform Performance group to Yes. The software caches only the directories needed for adding address attributes.

18.6.6.4.8 DSF2 walk sequencing

When you perform DSF2 walk sequencing in the software, the software adds delivery sequence information to your data, which you can use with presorting software to qualify for walk-sequence discounts.

→ Remember

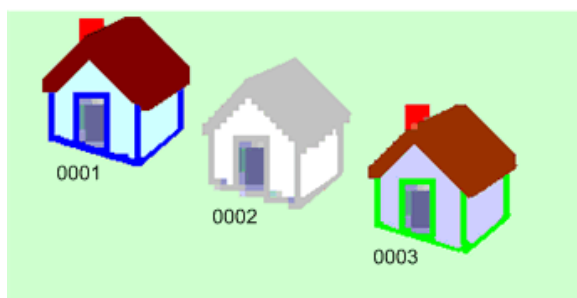
The software does not place your data in walk sequence order.

Include the DSF2 Walk Sequencer transform to enable walk sequencing.

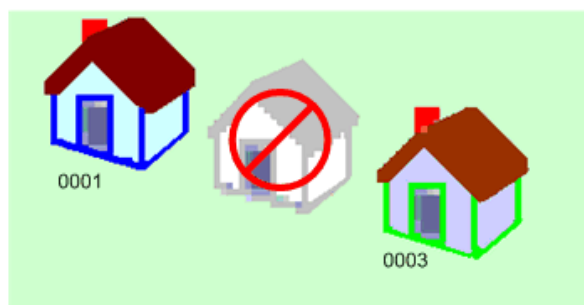
18.6.6.4.8.1 Pseudo sequencing

DSF2 walk sequencing is often called pseudo sequencing because it mimics USPS walk sequencing. Where USPS walk-sequence numbers cover every address, DSF2 processing provides pseudo sequence numbers for only the addresses in that particular file.

Walk sequencing includes every U.S. address, whether or not they're included in your list or mailing.



Pseudo sequencing includes only the addresses in your list. In the illustration below, assume the middle house is not included in your list.



The software uses DSF2 data to assign sequence numbers for all addresses that are DPV-confirmed delivery points (DPV_Status = Y). Other addresses present in your output file that are not valid DPV-confirmed delivery points (DPV_Status = S, N, or D) will receive "0000" as their sequence number. All other addresses will have a blank sequence number.

i Note

When you walk-sequence your mail with the software, remember the following points:

- (Batch only.) DSF2 walk sequencing is available only in batch mode. You cannot assign sequence numbers in real time.
- Reprocess if you have made file changes. If your data changes in any way, you must re-assign sequence numbers. Sequence numbers are valid only for the data file as you process it at the time.

18.6.6.4.9 Break key creation

Break keys create manageable groups of data. They are created when there are two or more fields to compare.

The DSF2 Walk Sequencer transform automatically forms break groups before it adds walk sequence information to your data. The software creates break groups based on the Postcode1 and Sortcode_Route fields.

Set options for how you want the software to configure the fields in the Data Collection Config group. Keeping the default settings optimizes the data flow and allows the software to make the break key consistent throughout the data.

Option	Default value
Replace NULL with space	Yes
Right pad with spaces	Yes

18.6.6.4.10 Enabling DSF2 walk sequencing

To enable DSF2 walk sequence, include the DSF2 Walk Sequencer transform in your data flow.

The input file for the DSF2 Walk Sequencer transform must have been pre-processed with CASS-certified software (such as the USA Regulatory Address Cleanse transform). To obtain an additional postage discount, include the DSF2_Business_Indicator output field information from CASS-certified software.

When you set up for DSF2 walk sequence processing, the following options in the USPS License Information group are required:

- Licensee Name
- DSF2 Licensee ID
- List ID

In addition to the required USPS License Information fields, make the following settings in the DSF2 Walk Sequencer transform:

1. Optional. Select Yes or No in the Common group, [Run as Separate Process](#) option. Select No if you are gathering DSF2 statistics. Select Yes to save processing time (if you don't need DSF2 statistics).
2. Enter the file path and file name (dsf.dir) to the Delivery Statistics directory in the [DelStats Directory](#) option in the Reference Files group. You may use the \$\$RefFilesAddressCleanse substitution parameter if you have it set up.
3. Enter the processing site location in the [Site Location](#) option of the Processing Options group. This is applicable only if you have more than one site location for DSF2 processing.

4. Make the following settings in the Data Collection Configuration group:
 - Select Yes or No in the *Replace Null With Space* option as desired.
 - Select Yes or No for the *Right Pad With Spaces* option as desired.
 - Select Yes or No for the *Pre Sorted Data* option (optional). We recommend that you keep the default setting of No so that Data Services sorts your data based on the break key fields (instead of using another software program).

18.6.6.4.11 DSF2 walk sequence input fields

Here is a list of the DSF2 walk sequence input fields.

i Note

These fields must have been output from CASS-certified software processing before they can be used as input for the DSF2 Walk Sequencer transform:

- Postcode1
- Postcode2
- Sortcode_Route
- LOT
- LOT_Order
- Delivery_Point
- DPV_Status
- DSF2_Business_Indicator (optional)

The software uses the information in these fields to determine the way the records should be ordered (walk sequenced) if they were used in a mailing list. The software doesn't physically change the order of your database. The software assigns walk-sequence numbers to each record based on the information it gathers from these input fields.

i Note

All fields are required except for the DSF2_Business_Indicator field.

The optional DSF2_Business_Indicator field helps the software determine if the record qualifies for saturation discounts. Saturation discounts are determined by the percentage of residential addresses in each carrier route. See the *USPS Domestic Mail Manual* for details about all aspects of business mailing and sorting discounts.

Related Information

[Reference Guide: USA Regulatory Address Cleanse, Regulatory Address Cleanse fields, Input fields](#)

18.6.6.4.12 DSF2 walk-sequence output fields

The software outputs walk-sequence number information to the following fields:

- Active_Del_Discount
- Residential_Sat_Discount
- Sortcode_Route_Discount
- Walk_Sequence_Discount
- Walk_Sequence_Number

Related Information

[Reference Guide: USA Regulatory Address Cleanse, Regulatory Address Cleanse fields, Output fields](#)

18.6.6.4.13 DSF2 reporting

There are reports and log files that the software generates for DSF2 augment and walk sequencing.

Find complete information about these reports and log files in the *Management Console Guide*.

Delivery Sequence Invoice Report

The USPS requires that you submit the Delivery Sequence Invoice report if you claim DSF2 walk-sequence discounts for this job.

US Addressing Report

- The US Addressing Report is generated by the USA Regulatory Address Cleanse transform.
- The Second Generation Delivery Sequence File Summary and Address Delivery Types sections of the US Addressing Report shows counts and percentages of addresses in your file that match the various DSF2 categories (if NCOALink is enabled). The information is listed for pre and post NCOALink processing.

DSF2 Augment Statistics Log File

The USPS requires that DSF2 licensees save information about their processing in the DSF2 log file. The USPS dictates the contents of the DSF2 log file and requires that you submit it to them monthly.

Log files are available to users with administrator or operator permissions.

Related Information

[Management Console Guide: Administrator, Administrator Management, Exporting certification logs, Exporting DSF2 certification log](#)

[Management Console Guide: Data Quality Reports, Descriptions of Reports, Delivery Sequence Invoice report](#)

[Management Console Guide: Data Quality reports, US Addressing Report](#)

18.6.6.4.13.1 DSF2 Augment Statistics Log File

If you create a DSF2 certified mailing list, submit the DSF2 Augment Statistics log file to the USPS.

The transform generates and stores the DSF2 Augment Statistics Log File in your repository. Export the log file using the Data Services Management Console.

The software uses the following naming convention for the log file: [DSF2_licensee_ID] [mm] [yy].dat

The USPS dictates the contents of the log file and requires that you submit it to them monthly by the third business day of each month by e-mail. For details about this process, read the *DSF2 Licensee Performance Requirements* document, which is available on the USPS PostalPro Web site at https://postalpro.usps.com/DSF2_LPR .

Log file retention and automatic deletion

The software deletes log files on a periodic basis. The default setting is 30 days. If 30 days is not a long enough period of time, we recommend that you extend the default setting to a longer period of time, such as 50 days. Additionally, the USPS requires that you keep USPS data around for them to examine for several years. Therefore, we recommend that you export the log files from your repository to a local directory that you create specifically for USPS data. Consider exporting the files on a monthly basis. An additional benefit to exporting the log files periodically is that you reduce the space in your repository for more efficient processing.

Extend the retention period for log files in the Central Management Console (CMC). Go to the Data Services Application Settings area. The option is [History Retention Period](#).

Make sure that you are familiar with the rules for retaining USPS information by checking with your DSF2 license agreement.

Related Information

[Setting the History Retention Period](#)

[USPS-required log files and reports](#)

18.6.6.5 NCOALink and Move Update

National Change of Address (NCOALink) is a part of USPS Move Update, which ensures that your mailing list contains up-to-date address information.

Move Updating is the process of checking your mailing list against the National Change of Address (NCOA) database. Move Updating ensures that your data is updated with current addresses. When you process for Move Update, the software updates records for individuals or businesses that have moved and have filed a Change of Address (COA) form with the USPS. Processing for Move Update qualifies your mailings for the discounts available for First-Class presorted mail.

In addition to supporting NCOALink, the USA Regulatory Address Cleanse transform supports other programs that are a part of Move Update: ANKLink and SuiteLink.

i Note

Mover ID is the name under which SAP Data ServicesData Quality Management SDK is certified with the USPS for NCOALink.

Parent topic: [Beyond the basic address cleansing \[page 533\]](#)

Related Information

[USPS DPV \[page 534\]](#)

[LACSLink \[page 557\]](#)

[SuiteLink \[page 567\]](#)

[USPS DSF2* \[page 570\]](#)

[USPS eLOT \[page 598\]](#)

[Early Warning System \(EWS\) \[page 599\]](#)

[USPS RDI \[page 601\]](#)

[GeoCensus \(USA Regulatory Address Cleanse \) \[page 604\]](#)

[Z4Change \[page 608\]](#)

[Suggestion lists \[page 610\]](#)

[Multiple data source statistics reporting \[page 613\]](#)

[SuiteLink \[page 567\]](#)

18.6.6.5.1 The importance of move updating

Processing your mailing lists with Move Update is a good business practice plus you receive postal discounts for your mailing.

The USPS requires move updating on all First Class presorted mailings. To help mailers meet this requirement, the USPS offers certain options, including NCOALink.

More than 40 million people and businesses move every year. Because most businesses and individuals communicate new address information to the USPS, it makes sense to use the USPS data to keep your records up-to-date. As the USPS expands move-updating requirements and more strictly enforces the existing regulations, move updating becomes increasingly more important.

With NCOALink, you benefit in the following ways:

- Improve mail deliverability
- Reduce the cost and time to forward mail
- Meet the USPS move-updating requirement for presorted First Class mail
- Prepare for the possible expansion of move-update requirements

Related Information

18.6.6.5.2 How NCOALink works

To understand the effects of NCOALink on your mailing lists, it is important to understand how the software uses NCOALink.

With Move Update enabled in your job setup, the software performs the following tasks:

1. The USA Regulatory Address Cleanse transform standardizes the input addresses. NCOALink requires parsed, standardized address data as input.
2. The transform searches the NCOALink database for records that match your parsed, standardized records.
3. If the transform finds a match, it updates the record with the move information from NCOALink, including the new address when available.
4. The transform further assigns postal and other codes to the move-updated address.
5. The software includes the following information in the output file based on the field class you select for output: CORRECT, MOVE-UPDATED, or BEST:
 - **CORRECT:** Outputs the original address, completed and corrected using values found in the directories and based on the standardization options you set in the transform.
 - **MOVE-UPDATED:** Outputs the address components that have been updated with move-updated address data.

i Note

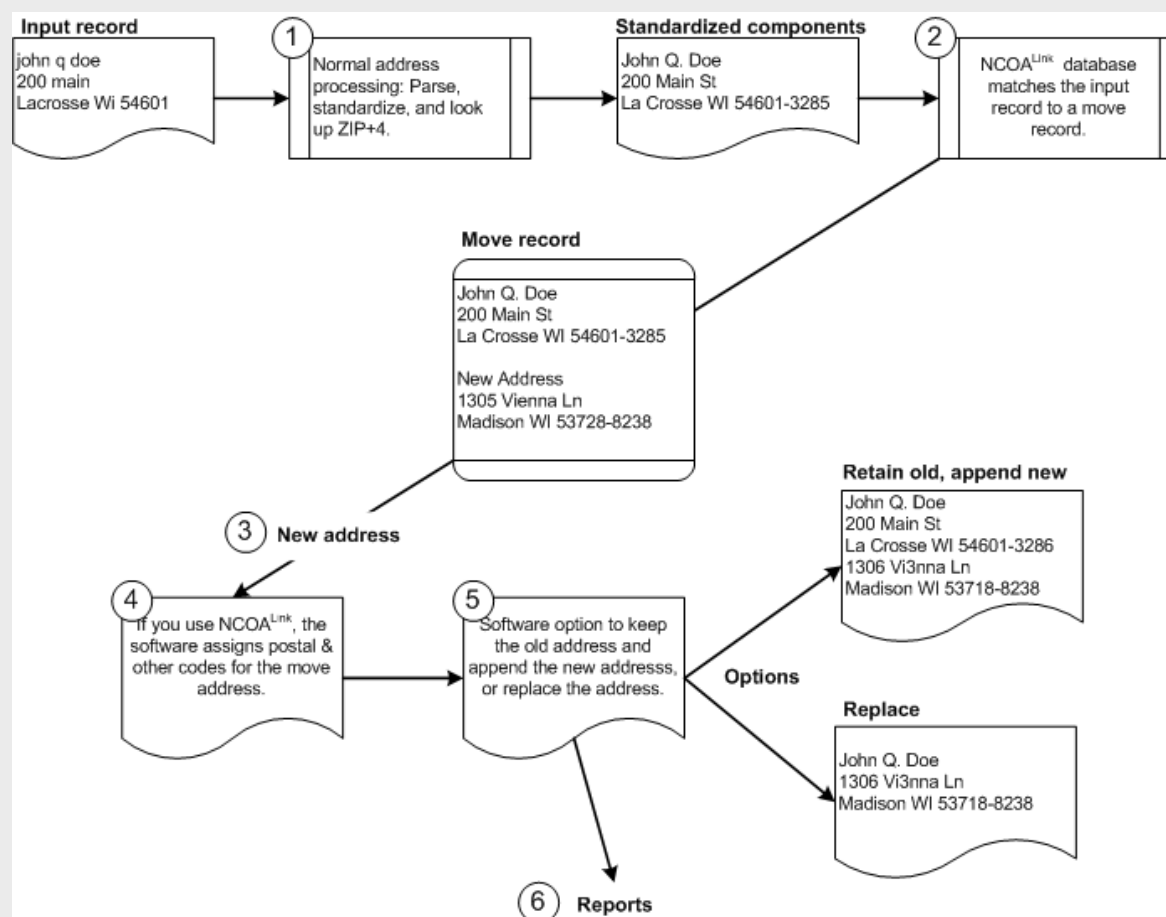
The transform looks for the move-updated address information in the U.S. National Directories. When the transform does not find the move-updated address in the U.S. National Directories, it populates the Move Updated fields with information from the Move Update Directories only. The transform does not update the Move Updated fields that are populated as a result of standardizing against the U.S. National Directories.

- **BEST:** Outputs the move-updated address when it exists and when it matches in the U.S. National directories. Or, output contains the original data when a move does not exist or when the move does not match in the U.S. National Directories.

Based on the [Apply Move to Standardized Fields](#) option in the NCOALink group, standardized components can contain either original or move-updated addresses.

- The software produces the reports and log files required for USPS compliance.

❖ Example



The following descriptions correspond to the circled numbers in the image:

- NCOALink requires parsed, standardized address data as input. Therefore, before NCOALink processing, the transform performs its normal processing on the address data.
- The transform searches the NCOALink database for a record that matches the parsed, standardized record.
- The transform receives the move information, including the new address if one is available.
- The transform looks up the move record that comes back from the NCOALink database, to assign postal and other codes.
- At your option, the transform can either retain the old address and append the new, or replace the old address with the new.
- The transform produces the reports and log files that the USPS requires for compliance.

18.6.6.5.3 NCOALink provider levels

NCOALink users fall into one of three categories of providers.

Specify the service provider level in the in the [Provider Level](#) option under the USPS License Information group.

i Note

Only provider levels supported by your registered keycodes display in the selection list.

Provider Level	Description
FSP (Full Service Provider)	Provides NCOALink processing to third parties.
LSP (Limited Service Provider)	Provides NCOALink processing to third parties and internally.
EUM (End User Mailer)	Provides NCOALink processing to in-house lists only.

18.6.6.5.4 NCOALink brokers and list administrators

An NCOALink user may have a broker or list administrator who owns the lists they are processing.

When there is a broker or list administrator involved, add contact information in the NCOALink group under [► Contact Detail list ► Contact Details ►](#).

The following table describes the broker and list administrator roles.

Role	Description
Broker	Directs business to an NCOALink service provider.
List Administrator	Maintains and stores lists.

Brokers and list administrators are different in two ways:

- List administrators don't send move-updated files back to the list owner.
- List administrators may have an NCOALink license.

If a list administrator, a broker, or both are involved in your job, ensure that you duplicate the Contact Detail group and complete the options for each of them separately.

18.6.6.5.5 Address not known (ANKLink)

Enhance NCOALink data with an additional 30 months of data when you purchase ANKLink.

Undeliverable-as-addressed (UAA) mail costs the mailing industry and the USPS a lot of money each year. The software provides NCOALink as an additional solution to UAA mail. With NCOALink, you also can have access to the USPS ANKLink data.

NCOALink limited service providers and end users receive change of address data for the preceding 18 months. The ANKLink option enhances that information by providing extra data about moves that occurred in the previous months 19 through 48.

i Note

The additional 30 months of data that comes with ANKLink indicate only that a move occurred and the date of the move; it does not provide the new address.

The ANKLink data helps you make informed choices regarding a contact. If the data indicates that the contact has moved, you can choose to suppress that contact from the list or try to acquire the new address from an NCOALINK full service provider.

If an ANKLink match exists, the software outputs a code in the [ANKLink_Return_Code](#) field and in the NCOALink Processing Summary report.

→ Tip

If you are an NCOALink full service provider, you already have access to the full 48 months of move data, including the new addresses.

18.6.6.5.5.1 ANKLink data

Request ANKLink data from the USPS National Customer Support Center (NCSC).

To request ANKLink data, either call the NCSC at 1-800-589-5766, or e-mail the NCSC at ncoalink@usps.gov. ANKLink data is not available from SAP.

The software detects if you're using ANKLink data. Therefore, you do not have to specify whether you're using ANKLink in your job setup.

18.6.6.5.5.2 ANKLink support for NCOALink provider levels

The software supports the three NCOALink provider levels defined by the USPS.

Software options vary by provider level and are activated based on the software package that you purchased. The following table shows the provider levels and support.

Provider Level	Service to Third Parties	COA Data Months	Data From USPS	ANKLink Support
FSP (Full Service Provider)	Yes. Requires at least 51% of services be for third-party processing.	48	Weekly	No
LSP (Limited Service Provider)	Yes. Can use services internally and for third parties.	18	Weekly	Yes

Provider Level	Service to Third Parties	COA Data Months	Data From USPS	ANKLink Support
End User	No	18	Monthly	Yes

→ Tip

If you are an NCOALink end user, you may complete a Stop Processing Alternative application and enter into an agreement with the USPS. After USPS approval, purchase the stop processing alternative functionality from SAP, which allows DPV and LACSLink processing to continue after a false positive address record is detected.

Related Information

[Stop Processing Alternative \[page 538\]](#)

18.6.6.5.6 Software performance

Processing speed is slower with NCOALink enabled.

In our tests, the software ran slower with NCOALink enabled than with it disabled. Your processing speed depends on the computer running the software and the percentage of input records affected by a move. For example, more moves equals slower performance.

Related Information

[Improving NCOALink processing performance \[page 591\]](#)

18.6.6.5.7 Getting started with NCOALink

Prepare for using NCOALink by completing USPS applications and configuring the software for NCOALink processing.

Before you begin NCOALink processing you need to perform the following tasks:

1. Complete the USPS certification process, and obtain approval to become an NCOALink service provider or end user.
2. Learn about the available output strategies and performance optimization options.
3. Configure the USA Regulatory Address Cleanse transform for NCOALink processing.
4. Include the transform in a job to produce NCOALink output.

18.6.6.5.8 What to expect from the USPS and SAP

Understand the expectations and roles between you, the USPS, and vendors when you become an NCOALink licensee.

NCOALink, and the license requirements that go with it, has created a new dimension in the relationship among mailers (you), the USPS, and vendors. It's important to be clear about what to expect from everyone.

18.6.6.5.8.1 Move updating is a business decision

Because NCOALink offers options to replace individual or family addresses with new addresses, service providers select whether to accept move updates related to either family or individual moves.

The USPS recommends that you make these choices only after careful thought about your customer relationships. Consider the following examples:

❖ Example

You mail monthly account statements, checks, and other correspondence to your customers. You have a fiduciary responsibility for ensuring your customers receive the correct information in a timely manner. The USPS recommends that you verify each move by sending a double postcard, or other easy-reply piece before you change a financial record to a new address. As technology changes, keep in touch with the USPS to determine the best way to verify address changes with your customers.

❖ Example

You have a business relationship with a customer who is married. You do not have a business relationship with the spouse. Ensure that you handle move updates carefully with respect to divorce or separation. Consider taking the time to confirm each move before you permanently update a record.

18.6.6.5.8.2 NCOALink security requirements

The USPS imposes strict security procedures for software vendors who use NCOALink data and provide NCOALink processing.

NCOALink data is confidential and sensitive. The USPS requires security procedures to prevent users from abusing the information.

The USPS requires that each mailing list contain at least 100 unique records. Therefore the USA Regulatory Address Cleanse transform checks your input files for at least 100 unique records. These checks make verification take longer, but, because it is a USPS requirement, the software must perform the check for each mailing list.

If the software finds that your data does not have 100 unique records, it issues an error and discontinues processing.

The process of checking for 100 unique records is a pre-processing step. So if the software does not find 100 unique records, it does not generate statistics and it does not perform any processing on the mailing list.

Related Information

[Getting started with NCOALink \[page 586\]](#)

18.6.6.5.8.2.1 How the software checks for 100 unique records

To satisfy the USPS requirement for 100 unique records for NCOALink processing, the software checks the input data for 100 unique records before it performs any processing on the data.

If the software finds 100 unique records, it continues and processes the job as usual. However, if the software does not find 100 unique records, it issues an error stating that your input data does not have 100 unique records. Also, if your data has fewer than 100 records, the software issues an error stating that there are not enough records to determine uniqueness.

The software concatenates all input fields of each record in the same order as the schema mapping in the software. For a record to be considered not unique, each field of a record must be identical to another record.

❖ Example

Comparing records

The following table shows how the software concatenates the fields in each record and determines non-unique records. The first and last rows in this example are not unique.

332	FRONT	STREET	NORTH	LACROSSE	WI	54601
332	FRONT	STREET	SOUTH	LACROSSE	WI	54601
331	FRONT	STREET	SOUTH	LACROSSE	WI	54601
332	FRONT	STREET	NORTH	LACROSSE	WI	54601

18.6.6.5.8.2.2 Finding unique records in multiple threads

When your NCOALink job is set up for multiple threads, the software checks each thread for 100 unique records.

The thread check doesn't necessarily check records in order. For example, thread #3 could be checked before thread #1. Therefore, each time you run the same multi-threaded job, there may be different unique record results.

You may receive an error even when you have 100 unique records per thread. An error can happen if one of the threads contains fewer than 100 records, and it is checked before other threads that contain at least 100 records. Consider reducing the number of threads before you try to process the job again.

Low volume of data and DOP > 1

When an NCOALink job is set up with the DOP (Degree of Parallelism) greater than 1, the software checks each thread for unique records within the first collection it processes. The software then shares knowledge of the unique records it finds with all other threads. When the software finishes processing the collection from the first thread, it counts the unique records found by all threads up to that point in time. It decides whether a thread has met the 100 record minimum. That thread may not necessarily be thread 1.

❖ Example

Your list has 3,050 records and you have the DOP set to 4. If the number of records per collection is 1000, each thread has a collection of 1000 records, except for the last thread, which has only 50 records. The software is likely to finish processing the thread with 50 records sooner than others, and it may make the pass—fail decision before it encounters 100 unique records. If you lower the DOP to 3, you may be able to run this job successfully.

18.6.6.5.9 About the NCOALink daily delete file

The USPS requires that NCOALink service provider licensees download the daily delete file each day before they perform NCOALink processing.

The daily delete file contains records that are pending deletion from the NCOALink data.

❖ Example

Jane Doe files a change of address with the USPS, but then she does not move. Even though her address is in the NCOALink directories as a COA (Change of Address), the daily delete file informs the mailer that she did not move.

The daily delete file informs mailers in the interim, until the USPS updates the record in the regular NCOALink directory.

Download the daily delete file and install it in the same directory as your NCOALink directories.

i Note

If you are an end user, you need the daily delete file only for processing Stage I or II test files. It is not required for normal NCOALink processing.

The following list contains important information about the daily delete file:

- The software fails verification when you are running an NCOALink certification stage test and you do not have the daily delete file installed.
- The USA Regulatory Address Cleanse transform supports only the ASCII version of the daily delete file.
- Do not rename the daily delete file from the original file name of `dailydel.dat`.
- The software issues a verification warning when the daily delete file is more than three days old.

18.6.6.5.9.1 Installing the NCOALink daily delete file

Install the NCOALink daily delete file each day that you plan to process mailing lists using NCOALink.

To download and install the NCOALink daily delete file, perform the following steps:

1. Go to the USPS Electronic Product Fulfillment site at <https://epf.usps.gov/> .
2. Find the Application Access section and log in to the USPS Electronic Product Fulfillment site. Use the e-mail and password that you established in the USPS Web Access Request form.
3. Download the NCOALink Daily Delete [TEXT] file to a location where you can extract the .tar file.

If your computer browser has pop-up blockers enabled, set your browser to accept popups.

4. Extract the `dailyDeletes_txt.tar` file.
5. Copy the `dailydel.dat` file to the same location as your NCOALink directories.

18.6.6.5.10 Output file strategies

By selecting specific Field Class Selections, configure your output file to meet your needs.

Select from the following Field Class Selections when you set up your output fields for NCOALink processing:

- **CORRECT:** Outputs the original address, completed and corrected using values found in the directories and based on the standardization options you set in the transform.
- **MOVE-UPDATED:** Outputs the address components that have been updated with move-updated address data.

i Note

The transform looks for the move-updated address information in the U.S. National Directories. When the transform does not find the move-updated address in the U.S. National Directories, it populates the Move Updated fields with information from the Move Update Directories only. The transform does not update the Move Updated fields that are populated as a result of standardizing against the U.S. National Directories.

- **BEST:** Outputs the move-updated address when it exists and when it matches in the U.S. National directories. Or, output contains the original data when a move does not exist or when the move does not match in the U.S. National Directories.

Based on the *Apply Move to Standardized Fields* option setting in the NCOALink group, standardized components can contain either original or move-updated addresses.

By default the output option *Apply Move to Standardized Fields* is set to *Yes* and the software updates standardized fields to contain details about the updated address available through NCOALink.

If you want to retain the old addresses in the standardized components and append the new ones to the output file, change the *Apply Move to Standardized Fields* option to *No*. Then you can use output fields such as *NCOALink_Return_Code* to determine whether a move occurred.

One way to set up your output file is to replicate the input file format, then append extra fields for move data. In the output records not affected by a move, most of the appended fields are blank. Alternatively, you can create a second output file specifically for move records. The following two approaches are possible:

- Output each record once, placing move records in the second output file and all other records in the main output file.
- Output move records twice; once to the main output file, and a second time to the second output file.

Both of these approaches require that you use an output filter to determine whether a record is a move-updated record.

18.6.6.5.11 Improving NCOALink processing performance

Many factors affect performance when processing NCOALink data.

One of the most critical factors is the volume of disk access that occurs. Often the most effective way to reduce disk access is to have sufficient memory available to cache data. Other critical factors that affect performance include hard drive speed, seek time, and the sustained transfer rate. When the time spent on disk access is minimized, the performance of the CPU becomes significant.

Related Information

[Finding unique records in multiple threads \[page 588\]](#)

18.6.6.5.11.1 Operating systems and processors

The computation involved in most of the software and NCOALink processing is very well-suited to the microprocessors found in most computers.

Examples include microcomputers made by Intel and AMD. RISC style processors like the processors in most UNIX systems are generally slower for this type of computation. A common PC can often run a single job through the software and NCOALink about twice as fast as a common UNIX system.

If you're looking for a cost-effective way of processing single jobs, a Windows server or a fast workstation can produce excellent results. Most UNIX systems have multiple processors and are at their best when they process several jobs at once.

To maximize the processor or core usage on your system, consider increasing the degree of parallelism (DOP) in the data flow properties. Increasing the DOP depends on the complexity of the data flow.

18.6.6.5.11.2 Memory

Obtain as much memory as possible for NCOALink processing.

NCOALink processing uses many gigabytes of data. The exact amount depends on your service provider level, the data format, and the specific release of the data from the USPS.

In general, if performance is critical, and especially if you are an NCOALink full service provider and you frequently run very large jobs with millions of records, you should obtain as much memory as possible. You may want to go as far as caching the entire NCOALink data set. You should be able to cache the entire NCOALink data set using 20 GB of RAM, with enough memory left for the operating system.

18.6.6.5.11.3 Data storage

If possible, dedicate the hard drive for NCOALink data entirely to the NCOALink process, at least while your job is running.

Any other process that competes for the same physical disk drive greatly reduces NCOALink performance.

Explore the possibility of using a RAID system (redundant array of independent discs) to achieve even higher transfer rates.

When the software accesses NCOALink data directly rather than from a cache, the most significant hard drive feature is the average seek time.

18.6.6.5.11.4 Data format

To increase performance, consider your data format.

The software supports both hash and flat file versions of NCOALink data. If you can cache the entire hash file set, the hash format may provide the best performance. Because the flat file data is significantly smaller, you can cache a larger share in a given amount of RAM. However, accessing the flat file data involves binary searches, which are slightly more time consuming than the direct access used with the hash file format.

18.6.6.5.11.5 Memory usage

Obtain as much memory as possible for NCOALink processing.

NCOALink processing uses many gigabytes of data. The exact amount depends on your service provider level, the data format, and the specific release of the data from the USPS.

In general, if performance is critical, obtain as much memory as possible. Your system needs a large amount of memory especially when you are an NCOALink full service provider and you frequently run very large jobs with millions of records. You may want to cache the entire NCOALink data set. Ensure that you leave enough memory for the operating system.

You can manually adjust the amount of memory to use by setting the *NCOALink Caching Mode* option to *Manual*. Or set the *NCOALink Caching Mode* option to *Auto* to have the software use available memory for caching.

18.6.6.5.11.6 Performance tips

Many factors can increase or decrease NCOALink processing speed.

Some performance factors are within your control and others may be inherent to your business. The following table contains some performance factors and discussion points to consider.

Factor	Description
Cache size	Using too little memory for NCOALink caching can cause unnecessary random file access and time-consuming hard drive seeks. Using too much memory can cause large files to be read from the disk into the cache even when only a tiny fraction of the data is used. The amount of cache that works best in your environment may require some testing to see what works best for your configuration and typical job size.
Directory location	Best practice is to put NCOALink directories on a local solid-state drive or a virtual RAM drive. Using a local solid-state drive or virtual RAM drive eliminates all I/O during NCOALink processing. If you have NCOALink directories on a hard drive, best practice is to use a defragmented local hard drive. Ensure that the hard drive is not accessed for anything other than the NCOALink data while you are running your job.
Match rate	The more records you process that have forwardable moves, the slower your processing is. Retrieving and decoding the new addresses takes time, so updating a mailing list regularly improves the processing speed on that list.
Input format	Best practice is to input data to the transform with discrete fields for the pre, first, middle, last, and post name fields. If your input has only a name line, the transform takes time to parse the line before checking NCOALink data.
File size	The transform processes larger files faster than smaller files. There is overhead when processing any job, but if a job includes millions of records, a few seconds of overhead is insignificant.

18.6.6.5.12 Enabling NCOALink processing

You must have access to the following files:

- NCOALink directories
- Current version of the USPS daily delete file
- DPV data files
- LACSLink data files

If you use a copy of the sample transform configuration, USARegulatoryNCOALink_AddressCleanse, NCOALink, DPV, and LACSLink are already enabled.

1. Open the [USA Regulatory Address Cleanse](#) transform and open the [Options](#) tab.
2. Set values for the options as appropriate for your situation.

For more information about the USA Regulatory Address Cleanse transform fields, see the [Reference Guide](#). The following table shows fields that are required only for specific provider levels.

Option group	Option name or subgroup	End user with- out stop proc- essing alterna- tive agreement	End user with stop processing alternative agreement	Full or limited service provider
USPS License Infor- mation	Licensee Name	yes	yes	yes
	List Owner NAICS Code	yes	yes	yes
	List ID	no	no	yes
	Customer Company Name	no	yes	yes
	Customer Company Address	no	yes	yes
	Customer Company Locality	no	yes	yes
	Customer Company Region	no	yes	yes
	Customer Company Postcode1	no	yes	yes
	Customer Company Postcode2	no	yes	yes
	Customer Company Phone	no	no	no
	List Processing Frequency	yes	yes	yes
	List Received Date	no	no	yes
	List Return Date	no	no	yes
	Provider Level	yes	yes	yes
NCOALink	PAF Details subgroup	no	no	All options are required, except Customer Parent Company Name and Customer Alternate Company Name.
	Service Provider Options subgroup	no	no	All options are required, except Buyer Company Name and Postcode for Mail Entry.

→ Tip

If you are a service provider and you need to provide contact details for multiple brokers, expand the NCOALink group, right-click [Contact Details](#) and click [Duplicate Option](#). An additional group of contact detail fields will be added below the original group.

Related Information

[Reference Guide: Transforms, Data Quality transforms, USA Regulatory Address Cleanse transform](#)
[Reference Guide: Transforms, Data Quality transforms, Address Cleanse reference, USPS certifications, NCOALink certification, About NCOA directories](#)
[About the NCOALink daily delete file \[page 589\]](#)
[Output file strategies \[page 590\]](#)
[Stop Processing Alternative \[page 538\]](#)

18.6.6.5.13 NCOALink log files

Use information from the StatisticsHandler to generate the USPS-required NCOALink log files.

The software automatically generates the USPS-required log files and names them according to USPS requirements. The software creates one log file per license ID. At the beginning of each month, the software starts new log files. The transform appends each log file with information about every NCOALink job processed that month for that specific license ID.

The software saves the log files to the repository where you can export them by using the Management Console.

The USPS requires you to save the NCOALink log files for five years.

The software produces the following move-related log files:

- CSL (Customer Service log)
- PAF (Processing Acknowledgement Form) Customer Information log
- BALA (Broker/Agent/List Administrator) log

The PAF Customer Information log and the BALA log are not required for end users.

The following table shows the log files required for Limited or Full Service Providers:

Log file	Required for	Description
CSL	• End users • Limited Service Providers • Full Service Providers	Contains one record for each list that you process. Each record details the results of change-of-address processing.

Log file	Required for	Description
PAF Customer Information	- Limited Service Providers - Full Service Providers	Contains the information that you provide for the PAF. Lists each unique PAF entry. If a list is processed with the same PAF information, the information appears once in the log file. When contact information for the list administrator changes, the software writes the information for both the list administrator and the corresponding broker to the PAF log file.
BALA	- Limited Service Providers - Full Service Providers	Contains the contact information that you entered for the broker or list administrator. Lists information for each broker or list administrator once. The USPS requires the Broker/Agent/List Administrator log file from service providers, even in jobs that do not involve a broker or list administrator. The software produces this log file for every job when you're a certified service provider.

Related Information

[Management Console Guide: NCOALink Processing Summary Report](#)

[Management Console Guide: Exporting NCOALink certification logs](#)

[Administrator Guide: Server Management, Setting the history retention period](#)

[Administrator Guide: Server Management, Setting the history retention period, USPS-required log files and reports](#)

18.6.6.5.13.1 Log file retention and deletion

The USPS requires NCOALink users to retain the log files for several years, but the software deletes the log files after a short period of time.

As an NCOALink provider, the USPS requires you to submit NCOALink log files every month. However, the software deletes log files on a periodic basis after 30 days by default. You can change the period of time by setting the [History Retention Period](#) option in the Data Services Application Settings group in the Central

Management Console (CMC). To avoid losing monthly log information, set the option to more than 31 days; we recommend a setting of 50 days.

In addition to sending monthly log files to the USPS, you are required to have the data available for the USPS to examine for several years after you process the job. Make sure you are aware of current USPS rules for data retention by checking your USPS licensing agreement. To ensure that you retain all required reports and logs before the data is deleted from the repository, we recommend that you export the required reports and logs to a local folder on a monthly basis.

Exporting the log files to a local folder also prevents the repository contents from becoming so large that the export process “times out” because the volume of statistics retained.

18.6.6.5.13.2 Log file names

The software follows a specific USPS file-naming convention for NCOALink log files.

The software follows a USPS file-naming scheme for the following log files:

- Customer Service log
- PAF Customer Information log
- Broker/Agent/List Administrators log

The following table describes the naming scheme for NCOALink log files.

Character 1		Characters 2 -5	Character 6		Characters 7-8	Extension
Log type		Platform ID	Month		Year	
B	Broker log	Exactly four characters long	1	January	Two characters , for example 10 for 2010	.DAT
C	Customer service log		2	February		
P	PAF log		3	March		
			4	April		
			5	May		
			6	June		
			7	July		
			8	August		
			9	September		
			A	October		
			B	November		
			C	December		

❖ Example

P1234C10.DAT is a PAF Log file generated in December 2010 for a licensee with the ID 1234.

18.6.6.6 USPS eLOT

eLOT narrows the mail carrier delivery route walk sequence to the house (delivery point) level.

eLOT enables you to sort your mailings to a more precise level. eLOT is available for U.S. addresses in the USA Regulatory Address Cleanse transform only.

eLOT takes line of travel one step further. The original Line of Travel (LOT) product narrowed the mail carrier delivery route to the block face level (Postcode2 level) by discerning whether an address resided on the odd or even side of a street or thoroughfare.

Parent topic: [Beyond the basic address cleansing \[page 533\]](#)

Related Information

[USPS DPV \[page 534\]](#)

[LACSLink \[page 557\]](#)

[SuiteLink \[page 567\]](#)

[USPS DSF2* \[page 570\]](#)

[NCOALink and Move Update \[page 581\]](#)

[Early Warning System \(EWS\) \[page 599\]](#)

[USPS RDI \[page 601\]](#)

[GeoCensus \(USA Regulatory Address Cleanse \) \[page 604\]](#)

[Z4Change \[page 608\]](#)

[Suggestion lists \[page 610\]](#)

[Multiple data source statistics reporting \[page 613\]](#)

[Enabling eLOT \[page 598\]](#)

[Set up the reference files \[page 498\]](#)

18.6.6.6.1 Enabling eLOT

1. Open the USA Regulatory Address Cleanse transform.
2. Open the *Options* tab, expand the Assignment Options group, and select Yes for the *Enable eLOT* option.
3. In the *Reference Files* group, set the path for your eLOT directory.

You can use the substitution variable `$$RefFilesAddressCleanse` for this option if you have it set up.

18.6.6.7 Early Warning System (EWS)

EWS helps reduce the amount of misdirected mail caused when the USPS adds valid delivery points between national directory updates.

EWS is available for U.S. records in the USA Regulatory Address Cleanse transform only.

Parent topic: [Beyond the basic address cleansing \[page 533\]](#)

Related Information

[USPS DPV \[page 534\]](#)

[LACSLink \[page 557\]](#)

[SuiteLink \[page 567\]](#)

[USPS DSF2* \[page 570\]](#)

[NCOALink and Move Update \[page 581\]](#)

[USPS eLOT \[page 598\]](#)

[USPS RDI \[page 601\]](#)

[GeoCensus \(USA Regulatory Address Cleanse \) \[page 604\]](#)

[Z4Change \[page 608\]](#)

[Suggestion lists \[page 610\]](#)

[Multiple data source statistics reporting \[page 613\]](#)

18.6.6.7.1 Overview of EWS

The EWS feature is the solution to the problem of misdirected mail caused by valid delivery points that appear between national directory updates. For example, suppose that 300 Main Street is a valid address and that 300 Main Avenue does not exist. A mail piece addressed to 300 Main Avenue is assigned to 300 Main Street on the assumption that the sender is mistaken about the correct suffix.

Now consider that construction is completed on a house at 300 Main Avenue. The new owner signs up for utilities and mail, but it may take a couple of months before the delivery point is listed in the national directory. All the mail intended for the new house at 300 Main Avenue will be mis-directed to 300 Main Street until the delivery point is added to the national directory.

The EWS feature solves this problem by using an additional directory which informs CASS users of the existence of 300 Main Avenue long before it appears in the national directory. When using EWS processing, the previously mis-directed address now defaults to a 5-digit assignment.

18.6.6.7.2 Start with a sample transform configuration

If you want to use the USA Regulatory Address Cleanse transform with the EWS option turned on, it is best to start with the sample transform configuration for EWS processing named `USARegulatoryEWS_AddressCleanse`.

18.6.6.7.3 EWS directory

The EWS directory contains four months of rolling data, and the USPS updates it weekly.

Each week, the USPS adds new data and drops data from the previous week. The USPS then publishes the latest EWS data. Each Friday, SAP converts the data to an SAP unique format and posts the directory in a Zip file (`EWyyymmdd.zip`) on the SAP support portal.

For more information about directories, and for instructions to access directories, see the latest directory letters at https://help.sap.com/viewer/p/ADDRESSING_DIRECTORIES.

i Note

Ensure that you access the most recent version by checking the version in the upper right corner of the Help Portal.

18.6.6.7.4 Enabling EWS

EWS is already enabled when you use the software's EWS sample transform, `USARegulatoryEWS_AddressCleanse`. These steps show how to manually set EWS.

1. Open the USA Regulatory Address Cleanse transform.
2. Open the *Options* tab and expand the Assignment Options group.
3. Select Enable for the *Enable EWS* option.
4. Expand the Reference Files group and enter a path for the *EWS Directory* option, or use the substitution variable `$$RefFilesAddressCleanse` if you have it set up.

Related Information

[Early Warning System \(EWS\) \[page 599\]](#)

18.6.6.8 USPS RDI

RDI (Residential Delivery Indicator) determines whether a given address is for a residence or non-residence.

Use RDI when you process your data for CASS certification or when you process in a non-certified mode. In addition, RDI does not require that you process with DPV (Delivery Point Validation).

Benefits of RDI

Many parcel shipping companies charge more for parcels going to U.S. residential addresses. The USPS does not charge extra for residential parcel delivery. That is because the USPS has found that 91-% of U.S. addresses are residential. Therefore, they are motivated to encourage the use of RDI by parcel mailers.

When you use RDI to determine the number of residential addresses in your mailing list, you can decide whether to use a commercial shipping service or to ship your parcels with the USPS and save money.

Parent topic: [Beyond the basic address cleansing \[page 533\]](#)

Related Information

[USPS DPV \[page 534\]](#)

[LACSLink \[page 557\]](#)

[SuiteLink \[page 567\]](#)

[USPS DSF2* \[page 570\]](#)

[NCOALink and Move Update \[page 581\]](#)

[USPS eLOT \[page 598\]](#)

[Early Warning System \(EWS\) \[page 599\]](#)

[GeoCensus \(USA Regulatory Address Cleanse \) \[page 604\]](#)

[Z4Change \[page 608\]](#)

[Suggestion lists \[page 610\]](#)

[Multiple data source statistics reporting \[page 613\]](#)

18.6.6.8.1 Start with a sample transform

If you want to use the RDI feature with the USA Regulatory Address Cleanse transform, it is best to start with the sample transform configuration, `USARegulatoryRDI_AddressCleanse`.

Sample transforms are located in the Transforms tab of the Object Library. This sample is located under `USA_Regulatory_Address_Cleanse` transforms.

18.6.6.8.2 How RDI works

RDI uses the 11-digit postcode to determine that an address is residential.

Before you use the RDI feature, install the USPS-supplied RDI directories. Then enable RDI processing in the software. The 11-digit postcode contains the following elements:

- Postcode1
- Postcode2
- DPBC

i Note

The software can sometimes determine if an address is residential by using just the Postcode2.

The software indicates whether an address is for a residence in the output component, *RDI_Indicator*.

Perform the following steps to use the RDI feature in the USA Regulatory Address Cleanse transform:

1. Install the USPS-supplied directories.
2. Specify the directory location in the Reference Files section of the transform.
3. Enable RDI processing in the Assignment options section of the transform.
4. Run the job.

Related Information

[Enabling RDI \[page 603\]](#)

18.6.6.8.3 RDI directory files

SAP ships the RDI directory files with the U.S. National Directory update.

Follow installation instructions for the RDI directory files in the directory letter.

RDI requires the directories listed in the following table.

File	Description
<code>rts.hs11</code>	For 11-digit postcode lookups (Postcode2 plus DPBC). The software uses the <code>rts.hs11</code> file when an address contains an 11-digit postcode. RDI bases the residence determination on the delivery point.

File	Description
<code>rts.hs9</code>	For 9-digit postcode lookups (Postcode2). The <code>rts.hs9</code> file is based on ZIP+4 codes. RDI is possible only when the addresses for the ZIP+4 are for residences or for non-residences.

18.6.6.8.3.1 Specifying the RDI directory path

In the Reference Files group, specify the location of your RDI directories in the [RDI Path](#) option. If RDI processing is disabled, the software ignores the [RDI Path](#) setting.

18.6.6.8.4 Enabling RDI

If you use a copy of the `USARegulatoryRDI_AddressCleanse` sample transform in your data flow, RDI is already enabled. However, if you are starting from a USA Regulatory Address Cleanse transform, make sure you enable RDI and set the location for the following RDI directories: `rts.hs11` and `rts.hs9`.

1. Open the [USA Regulatory Address Cleanse](#) transform.
2. In the [Options](#) tab expand the Reference Files group, and enter the location of the RDI directories in the [RDI Path](#) option, or use the substitution variable `$$RefFilesAddressCleanse` if you have it set up.
3. Expand the Assignment Options group, and select [Yes](#) for the [Enable RDI](#) option.

18.6.6.8.5 RDI output field

The RDI feature outputs information to a single output field.

Include the RDI output field in your field setup. The software populates the RDI field only when the [Enable RDI](#) option in the Assignment Options group in the transform configuration is set to [Yes](#).

Field	Length	Description
RDI_Indicator	1	Populated with one of the following values: • Y: Address is for a residence • N: Address is not for a residence

18.6.6.8.6 RDI in reports

A few of the software's reports have additional information because of the RDI feature.

18.6.6.8.6.1 USPS Form 3553

The software uses specific RDI information for the USPS Form 3553 report, when you process in CASS certification mode.

The software populates the USPS Form 3553 with the number of residences. It also includes the information in the CASS header record.

18.6.6.8.6.2 Statistics files

The statistics file contains RDI counts and percentages.

18.6.6.9 GeoCensus (USA Regulatory Address Cleanse)

The GeoCensus option of the USA Regulatory Address Cleanse transform offers geographic and census coding for enhanced sales and marketing analysis. It is available for U.S. records only.

i Note

GeoCensus functionality in the USA Regulatory Address Cleanse transform will be deprecated in a future version. It is recommended that you upgrade any data flows that currently use the GeoCensus functionality to use the Geocoder transform. For instructions on upgrading from GeoCensus to the Geocoder transform, see the *Upgrade Guide*.

Parent topic: [Beyond the basic address cleansing \[page 533\]](#)

Related Information

[USPS DPV \[page 534\]](#)

[LACSLink \[page 557\]](#)

[SuiteLink \[page 567\]](#)

[USPS DSF2* \[page 570\]](#)

[NCOALink and Move Update \[page 581\]](#)

[USPS eLOT \[page 598\]](#)

[Early Warning System \(EWS\) \[page 599\]](#)

[USPS RDI \[page 601\]](#)

[Z4Change \[page 608\]](#)

[Suggestion lists \[page 610\]](#)

[Multiple data source statistics reporting \[page 613\]](#)

[How GeoCensus works \[page 605\]](#)

[GeoCensus directories \[page 606\]](#)

[Enabling GeoCensus coding \[page 607\]](#)

[Geocoding \[page 378\]](#)

18.6.6.9.1 How GeoCensus works

By using GeoCensus, the USA Regulatory Address Cleanse transform can append latitude, longitude, and census codes such as census tract and Metropolitan Statistical Area (MSA) to your records, based on ZIP+4 codes. MSA is an aggregation of US counties into Metropolitan Statistical Areas assigned by the US Office of Management and Budget. You can apply the GeoCensus codes during address standardization and postcode2 assignment for simple, “one-pass” processing.

The transform cannot, by itself, append demographic data to your records. The transform lays the foundation by giving you census coordinates via output fields. To append demographic information, you need a demographic database from another vendor. When you obtain one, we suggest that you use the matching process to match your records to the demographic database, and transfer the demographic information into your records. (You would use the MSA and census tract information as match criteria, then use the Best Record transform to post income and other information.)

Likewise, the transform does not draw maps. However, you can use the latitude and longitude assigned by the transform as input to third-party mapping applications. Those applications enable you to plot the locations of your customers and filter your database to cover a particular geographic area.

18.6.6.9.2 The software provides census coordinates

The software cannot, by itself, append demographic data to your records. The software simply lays the foundation by giving you census coordinates. To append demographic information, you need a demographic database from another vendor. When you get that, we suggest that you use our Match transform to match your records to the demographic database and transfer the demographic information into your records. (In technical terms, you would use the MSA and Census block/tract information as match fields, then use the Best Record post-match operation in the Match transform to transfer income and other information.

Likewise, the software does not draw maps. However, you can use the latitude and longitude assigned by the software as input to third-party mapping software. Those programs enable you to plot the locations of your customers and filter your database to cover a particular geographic area.

Related Information

[Best record \[page 458\]](#)

18.6.6.9.3 Get the most from the GeoCensus data

You can combine GeoCensus with the functionality of mapping software to view your geo-enhanced information. It will help your organization build its sales and marketing strategies. Here are some of the ways you can use the GeoCensus data, with or without mapping products.

Information type	How GeoCensus can help
Market analysis	You can use mapping applications to analyze market penetration, for instance. Companies striving to gain a clearer understanding of their markets employ market analysis. This way they can view sales, marketing, and demographic data on maps, charts, and graphs. The result is a more finely targeted marketing program. You will understand both where your customers are and the penetration you have achieved in your chosen markets.
Predictive modeling and target marketing	You can more accurately target your customers for direct response campaigns using geographic selections. Predictive modeling or other analytical techniques allow you to identify the characteristics of your ideal customer. This method incorporates demographic information used to enrich your customer database. From this analysis, it is possible to identify the best prospects for mailing or telemarketing programs.
Media planning	For better support of your advertising decisions, you may want to employ media planning. Coupling a visual display of key markets with a view of media outlets can help your organization make more strategic use of your advertising dollars.
Territory management	GeoCensus data provides a more accurate market picture for your organization. It can help you distribute territories and sales quotas more equitably.
Direct sales	Using GeoCensus data with market analysis tools and mapping software, you can track sales leads gathered from marketing activities.

18.6.6.9.4 GeoCensus directories

The path and file names for the following directories must be defined in the Reference Files option group of the USA Regulatory Address Cleanse transform before you can begin GeoCensus processing. You can use the substitution variable `$$RefFilesDataCleanse`.

Directory name	Description
ageo1-10	Address-level GeoCensus directories are required if you choose Address for the Geo Mode option under the Assignment Options group.
cgeo2.dir	Centriod-level GeoCensus directory is required if you choose Centroid for the Geo Mode option under the Assignment Options group.

18.6.6.9.5 GeoCensus mode options

To activate GeoCensus in the transform, you need to choose a mode in the [Geo Mode](#) option in the Assignment Options group.

Mode	Description
Address	Processes Address-level GeoCensus only.
Both	Attempts to make an Address-level GeoCensus assignment first. If no assignment is made, it attempts to make a Centroid-level GeoCensus assignment.
Centroid	Processes Centroid-level GeoCensus only.
None	Turns off GeoCensus processing.

18.6.6.9.6 GeoCensus output fields

You must include at least one of the following generated output fields in the USA Regulatory Address Cleanse transform if you plan to use the GeoCensus option:

- AGeo_CountyCode
- AGeo_Latitude
- AGeo_Longitude
- AGeo_MCDCode
- AGeo_PlaceCode
- AGeo_SectionCode
- AGeo_StateCode
- CGeo_BSACode
- CGeo_Latitude
- CGeo_Longitude
- CGeo_Metrocode
- CGeo_SectionCode

Find descriptions of these fields in the *Reference Guide*.

18.6.6.9.7 Sample transform configuration

To process with the GeoCensus feature in the USA Regulatory Address Cleanse transform, it is best to start with the sample transform configuration created for GeoCensus. Find the sample configuration, `USARegulatoryGeo_AddressCleanse`, under `USA_Regulatory_Address_Cleanse` in the Object Library.

18.6.6.9.8 Enabling GeoCensus coding

If you use a copy of the `USARegulatoryGeo_AddressCleanse` sample transform file in your data flow, GeoCensus is already enabled. However, if you are starting from a USA Regulatory Address Cleanse transform, make sure you define the directory location and define the [Geo Mode](#) option.

1. Open the [USA Regulatory Address Cleanse](#) transform.

2. In the *Options* tab, expand the *Reference Files* group.
3. Set the locations for the `cgeo.dir` and `ageo1-10.dir` directories based on the Geo Mode you choose.
4. Expand the Assignment Options group, and select either *Address*, *Centroid*, or *Both* for the *Geo Mode* option.
If you select *None*, the transform will not perform GeoCensus processing.

Related Information

[GeoCensus \(USA Regulatory Address Cleanse \) \[page 604\]](#)

18.6.6.10 Z4Change

The Z4Change feature enhances automation compatibility and improves deliverability.

With Z4Change, the software adds or updates ZIP+4 codes to addresses and updates incorrect ZIP+4 codes. The Z4Change option is available in the USA Regulatory Address Cleanse transform only.

Parent topic: [Beyond the basic address cleansing \[page 533\]](#)

Related Information

[USPS DPV \[page 534\]](#)

[LACSLink \[page 557\]](#)

[SuiteLink \[page 567\]](#)

[USPS DSF2* \[page 570\]](#)

[NCOALink and Move Update \[page 581\]](#)

[USPS eLOT \[page 598\]](#)

[Early Warning System \(EWS\) \[page 599\]](#)

[USPS RDI \[page 601\]](#)

[GeoCensus \(USA Regulatory Address Cleanse \) \[page 604\]](#)

[Suggestion lists \[page 610\]](#)

[Multiple data source statistics reporting \[page 613\]](#)

18.6.6.10.1 Enable Z4Change for faster processing

Enabling the Z4Change option can save processing time for large databases.

Z4Change is most cost-effective for databases that are large and fairly stable—for example, databases of regular customers, subscribers, and so on. In our tests, based on files in which five percent of records were affected by a ZIP+4 change, total batch processing time was one third the normal processing time.

When you use the transform interactively—that is, processing one address at a time—there is less benefit from using Z4Change.

18.6.6.10.2 Z4Change and USPS rules

The USPS requires that mailing lists undergo a complete assignment process every three years.

Use Z4Change only for updating a database that has previously been put through a full validation process.

18.6.6.10.3 Z4Change directory

The USPS updates the Z4Change directory each month.

Users who have paid extra for the Z4Change option, can access the Z4Change directory with the monthly USPS directory delivery.

The Z4Change directory contains a list of all the ZIP Codes and ZIP+4 codes in the country.

You define the location of the Z4Change directory in the Reference Files section of the USA Regulatory Address Cleanse transform.

18.6.6.10.4 Start with a sample transform

If you want to use the Z4Change feature in the USA Regulatory Address Cleanse transform, it is best to start with the sample transform, `USARegulatoryZ4Change_AddressCleanse`.

18.6.6.10.5 Enabling Z4Change

If you use a copy of the Z4Change transform configuration file sample (`USARegulatoryZ4Change_AddressCleanse`) in your data flow, Z4Change is already enabled. However, if you are starting from a USA Regulatory Address Cleanse transform, make sure you define the directory location and define the *Z4Change Mode* option.

1. Open the *USA Regulatory Address Cleanse* transform.

2. On the [Options](#) tab, expand the Reference Files group.
3. Set the location for the `z4change.dir` directory in the [Z4Change Directory](#) option.
4. Expand Z4Change options group and select [Yes](#) for the [Enable Z4Change](#) option.
5. In the Z4Change option group, enter the month and year that the input records were most recently ZIP+4 updated in the [Last ZIP+4 Assign Date](#) option.

18.6.6.11 Suggestion lists

Use Suggestion List processing in transactional projects with the USA Regulatory Address Cleanse, Global Address Cleanse, and the Global Suggestion List transforms.

Use suggestion lists to complete and populate addresses that have minimal data. Suggestion lists can offer suggestions for possible matches when the transform cannot find an exact match.

For information about suggestion lists for Global Address Cleanse and Global Suggestion lists transforms, see [Global Address Cleanse Suggestion List option \[page 527\]](#) and [Global Suggestion List transform \[page 528\]](#).

i Note

Suggestion list processing is not available for batch processing. In addition, if you have suggestion lists enabled, you are not eligible for CASS discounts, and the software does not produce the required CASS documentation.

Parent topic: [Beyond the basic address cleansing \[page 533\]](#)

Related Information

[USPS DPV \[page 534\]](#)

[LACSLink \[page 557\]](#)

[SuiteLink \[page 567\]](#)

[USPS DSF2® \[page 570\]](#)

[NCOALink and Move Update \[page 581\]](#)

[USPS eLOT \[page 598\]](#)

[Early Warning System \(EWS\) \[page 599\]](#)

[USPS RDI \[page 601\]](#)

[GeoCensus \(USA Regulatory Address Cleanse \) \[page 604\]](#)

[Z4Change \[page 608\]](#)

[Multiple data source statistics reporting \[page 613\]](#)

[Global Suggestion List transform \[page 528\]](#)

18.6.6.11.1 How suggestion lists work

The USA Regulatory Address Cleanse transform presents address suggestions when it cannot find an exact match to an address in the postal directories.

Ideally, when the transform looks up an address in the USPS postal directories, it finds exactly one matching record with a matching combination of locality, region, and postcode. Then, during the look-up in the USPS national ZIP+4 directory, the software should find exactly one record that matches the address. However, the transform doesn't always find an exact match.

Breaking ties

Sometimes it's impossible to pinpoint an input address to one matching record in the directory. At other times, the software may find several directory records that are near matches to the input data.

When the software is close to a match, but not quite close enough, it assembles a list of the near matches and presents them as suggestions. When you choose a suggestion, the software tries again to assign the address.

❖ Example

Incomplete last line

Given the incomplete last line below, the software could not reliably choose one of the four localities. But if you choose one, the software can proceed with the rest of the assignment process.

Input record	Possible matches in the City/ZCF directories
Line1 = 1000 vine Line2 = lacr wi	La Crosse, WI 54603 Lancaster, WI 53813 La Crosse, WI 54601 Larson, WI 54947

❖ Example

Missing directional

The same can happen with address lines. A common problem is a missing directional. In the following example, there is an equal chance that the directional could be North or South. The software has no basis for choosing one way or the other.

Input record	Possible matches in the ZIP+4 directory
Line1 = 615 losey blvd Line2 = 54603	600–699 Losey Blvd N 600–698 Losey Blvd S

❖ Example

Missing suffix

A missing suffix would cause similar behavior as with a missing directional. There is an equal chance that the suffix could be Pl or St. The software has no basis for choosing one or the other.

Input record	Possible matches in the ZIP+4 directory
Line1 = 121 dorn Line2 = 54601	100–199 Dorn Pl 101–199 Dorn St

❖ Example

Misspelled street names

A misspelled or incomplete street name can result in address suggestions.

Input record	Possible matches in the ZIP+4 directory
Line1 = 4100 marl Line2 = minneapolis mn	4100–4199 Marshall 55421 4100–4199 Maryland 55427

18.6.6.11.1.1 When you need more information

Sometimes you need more information before you select a suggestion.

When the software produces a suggestion list, you need some basis for selecting one of the possible matches. Maybe the person providing the address is with you, or you have another resource that contains the address for the customer.

❖ Example

The following list contains examples of additional information resources:

- Operators taking information over the phone can ask for more information from the customer to decide what suggestion to choose.
- Operators entering data from a consumer coupon that is a little smudged may be able to choose a suggestion based on the information that is not smudged.

If none of the other resources prove helpful, do not guess. The address can remain unassigned until you have better or more complete information.

18.6.6.11.1.2 CASS and the suggestion list feature

The software does not produce a USPS Form 3553 when you enable suggestion lists.

First of all, the USPS suspects that users may be tempted to guess when presented with suggestions. An incorrect guess may result in misrouted mail that is expensive for the USPS to handle. Second of all, the USPS Form 3553 is available only when you process in batch mode. The form is available only when you process in batch mode with the *Disable Certification* option set to *No*.

To produce a CASS-certified mailing using suggestion lists, perform two processes:

- Run the addresses through a real-time job with suggestions enabled.
- Run the addresses again through a batch process with *Disable Certification* set to *No*.

The batch job is CASS-compliant and the software generates a USPS Form 3553 that covers the entire mailing database.

18.6.6.11.1.3 Integrating functionality

Suggestion Lists functionality is designed to be integrated into your own custom applications via the Web Service. For information about integrating Data Services for web applications, see the *Integrator Guide*.

18.6.6.11.1.4 Sample suggestion lists blueprint

If you want to use the suggestion lists feature, it is best to start with one of the sample transforms configured for it. The sample transform is named `USARegulatorySuggestions_Address_Cleanse`. It is available for the USA Regulatory Address Cleanse transform.

18.6.6.12 Multiple data source statistics reporting

The USA Regulatory Address Cleanse transform reports statistics for a job that contains multiple data sources based on logical groupings.

An input database that contains a combination of several mailing lists uses unique list identifiers. The list identifiers track what list a record originates from. The unique list identifier can be a name or a number. The identifier resides in the same field across all of the combined lists.

When you map your input fields, map the field that contains the unique list ID to the software input field *Data_Source_ID*. The software collects statistics for each list using the *Data_Source_ID* input field. Some of the generated reports contain a summary for the entire list, and a separate summary per list based on the value mapped into the *Data_Source_ID* field.

! Restriction

For compliance with NCOALink reporting restrictions, the USA Regulatory Address Cleanse transform does not support processing multiple mailing lists associated with different PAFs (Processing Acknowledgment Forms). Therefore, for NCOALink processing, the software considers all records in the input file to be a single mailing list. And, the software reports the list as a single mailing list in the Customer Service Log (CSL) file.

! Restriction

The *Gather Statistics Per Data Source* functionality is not supported when the *Enable Parse Only* or *Enable Geo Only* options in the Non Certified Options group are set to *Yes*.

Parent topic: [Beyond the basic address cleansing \[page 533\]](#)

Related Information

[USPS DPV \[page 534\]](#)
[LACSLink \[page 557\]](#)
[SuiteLink \[page 567\]](#)
[USPS DSF2* \[page 570\]](#)
[NCOALink and Move Update \[page 581\]](#)
[USPS eLOT \[page 598\]](#)
[Early Warning System \(EWS\) \[page 599\]](#)
[USPS RDI \[page 601\]](#)
[GeoCensus \(USA Regulatory Address Cleanse \) \[page 604\]](#)
[Z4Change \[page 608\]](#)
[Suggestion lists \[page 610\]](#)
[Gathering statistics per list \[page 615\]](#)

18.6.6.12.1 Data_Source_ID field

The software tracks statistics for each list based on the [Data_Source_ID](#) input field.

❖ Example

There are five mailing lists combined into one list for input into the USA Regulatory Address Cleanse transform. Each list has a common field named [List_ID](#), and a unique identifier in the [List_ID](#) field: N, S, E, W, C. The following table shows the input mapping.

Transform input field name	Input schema column name	Type
Data_Source_ID	List_ID	varchar(80)

To obtain DPV statistics for each [List_ID](#), process the job and then open the US Addressing report.

The first DPV Summary section in the US Addressing report lists the Cumulative Summary, which reports the totals for the entire database. Subsequent DPV Summary sections list summaries per [Data_Source_ID](#). The example in the following table shows the counts and percentages for the entire database (cumulative summary) and for [Data_Source_ID](#) "N".

Statistic	DPV cumulative summary count	DPV summary for		
		%	Data_Source_ID “N”	%
DPV Validated Ad- dresses	1,968	3.94	214	4.28

Statistic	DPV cumulative summary count	DPV summary for % <i>Data_Source_ID</i> "N"		%
Addresses Not DPV Valid	3,032	6.06	286	5.72
CMRA Validated Addresses	3	0.01	0	0.00
DPV Vacant Addresses	109	0.22	10	0.20
DPV NoStats Addresses	162	0.32	17	0.34

Related Information

[Group statistics and cumulative statistics reports \[page 617\]](#)

18.6.6.12.2 Gathering statistics per list

Assign a field to uniquely identify each list so that the transform lists statistics separately per list.

Before setting up the USA Regulatory Address Cleanse transform to gather statistics per list, identify the field that uniquely identifies each list. For example, a mailing list that is comprised of more than one source might contain lists that have a field named LIST_ID that uniquely identifies each list.

1. Open the USA Regulatory Address Cleanse transform in the data flow and then click the [Options](#) tab.
2. Expand the Report and Analysis group and select Yes for the [Generate Report Data](#) and the [Gather Statistics Per Data Source](#) options.
3. Click the [Input](#) tab and click the [Input Schema Column Name](#) field next to the Data_Source_ID field for uniquely identifying a list.
A drop menu appears.
4. Click the drop menu and select the input field from your database that you've chosen as the common field for uniquely identifying a list. In the scenario above, that would be the LIST_ID field.
5. Continue with the remaining job setup tasks and execute your job.

18.6.6.12.3 Physical Source Field and Cumulative Summary

The Address Standardization, Address Information Code, and USA Regulatory Locking reports do not include a Cumulative Summary, but they present records sorted by the [Data_Source_ID](#) value.

Some reports include both of the following list types:

- A report per list based on the [Data_Source_ID](#) field, identified in the report footer by "Physical Source Field"

- A summary of the entire list, identified in the report footer by “Cumulative Summary”

i Note

When you enable NCOALink, the software reports a summary per list only for the following sections of the NCOALink Processing Summary Report:

- NCOALink Move Type Summary
- NCOALink Return Code Summary
- ANKLink Return Code Summary

Special circumstances

There are some circumstances when the words “Cumulative Summary” and “Physical Source Field” do not appear in the report footer sections:

- When the *Gather Statistics Per Data Source* option is set to *No*
- When the *Gather Statistics Per Data Source* option is set to *Yes* and there is only one *Data_Source_ID* value present in the list, but it is empty

18.6.6.12.3.1 USPS Form 3553 and group reporting

The USPS Form 3553 includes a summary of the entire list and a report per list based on the *Data_Source_ID* field.

❖ Example

Cumulative Summary

The USPS Form 3553 designates the summary for the entire list with the words “Cumulative Summary.” It appears in the footer of the report. In addition, the Cumulative Summary of the USPS Form 3553 contains the total number of lists in the job in Section B, under **Number of Lists**.

❖ Example

Physical Source Field

The USPS Form 3553 designates the summary for each Individual list with the words “Physical Source Field” followed by the *Data_Source_ID* value. It appears in the footer. The data in the report is for the list designated by the data source ID only.

18.6.6.12.3.2 Group statistics and cumulative statistics reports

There are various reports that include either a cumulative statistics section or a group statistics section, or both sections.

Cumulative statistics are summaries for the entire mailing list. Group statistics are based on the Physical Source Field.

The reports that include both cumulative and group statistics include the following:

- Address Validation Summary
- Address Type Summary
- US Addressing

The reports that include only a group statistics section but not a cumulative summary include the following:

- Address Information Code Summary
- Address Standardization
- US Regulatory Locking

18.6.7 Data quality statistics repository tables

You can generate data quality statistics for the Data Cleanse, Geocoder, and Global Address Cleanse transforms.

When you enable the data quality statistics settings in the transform and run your job, the software sends statistics to tables in the repository. Based on the table you choose to generate, the statistics contain summary and non-summary information about the cleansing and assignment processes.

There are two main reasons for generating and using the data quality statistics:

- To view aggregated results and summary information about the quality of your data.
- To drill into specific records that the software identified as low quality.

Enable data quality statistics tables in the applicable transform by making settings in the [Data Quality Statistics Settings](#) group under [Report and Analysis](#).

Related Information

[Cleansing Package Builder \[page 348\]](#)

[Geocoding \[page 378\]](#)

[Cleansing global address data \[page 493\]](#)

[Reference Guide: Repository statistics tables for data quality](#)

18.6.8 Data Quality support for native data types

The Data Quality transforms process most incoming data types as character data, with some exceptions.

In the mapping process, you set a data type. If you map a noncharacter data type on input, the transform converts the data to a character string before passing it through the Data Quality transform for processing.

There are some data types that the software recognizes and processes using the same data type as on input.

❖ Example

You map a date type field to a Data Quality date type input field.

When you map your input fields using specific Data Quality data type fields, the software recognizes and sorts the incoming data using the specific data type. Mapping applicable input fields to Data Quality data type fields increases efficiency because there are fewer fields the transform has to convert to character data.

Related Information

[Data Types](#)

18.6.9 Data Quality data type definitions

The Data Quality transforms have four field attributes to define the field.

The four field attributes include the following:

- Name
- Type
- Length
- Scale

These attributes are listed in the Input and output tab of the transform editor.

In the Input tab, the attribute Name is listed under the *Transform Input Field Name* column. The Type, Length, and Scale attributes are listed under the *Type* column in the format <type>(<length>, <scale>).

The Output tab also contains the four field attributes listed above. The attribute Name is listed under the *Field_Name* column. The Type, Length, and Scale attributes are listed under the *Type* column in the format <type>(<length>, <scale>).

18.6.10 Data Quality support for NULL values

The Data Quality transforms process NULL values as NULL unless there is data available to populate the field on output.

A field that is NULL is passed through processing with the NULL marker preserved unless there is data available to populate the field on output. When there is data available, the field is output with the data available

instead of NULL. The benefit of this treatment of NULL is that the Data Quality transforms treat a NULL marker as unknown instead of empty.

i Note

If all fields of a record contain NULL, the transform will not process the record, and the record will not be a part of statistics and reports.

Related Information

[Reference Guide: Scripting language, Language syntax, NULL values, NULL values and empty strings \[page 809\]](#)

18.7 Query transform

Use the Query transform anywhere in a data flow to perform SQL type operations on source data such as grouping, joining, and ordering.

The Query transform serves many purposes based on the situation. Many of the examples we use throughout the Data Services documentation involve using a Query transform. For example, see how to use the Query transform to flatten a nested schema in the Tutorial located at https://help.sap.com/viewer/p/SAP_DATA_SERVICES.

Perform the following operations with the Query transform:

- Ensure that the output contains only distinct rows and no duplicates
- Construct inner and left outer joins using multiple input schemas, and include conditions and ranking
- Use joins proposed by the transform
- Output specific data based on filters or manual selection
- Map columns from input to output schemas
- Remap schemas when you change the data coming into the Query transform
- Perform transformations and functions on data
- Perform data nesting and unnesting
- Add new columns, nested schemas, and function results to the output schema
- Assign primary keys to output columns

[Schema In and Schema Out \[page 620\]](#)

The *Schema In* and *Schema Out* panes determine the fields that continue on in your data flow as well as field mapping.

[Query transform input schema \[page 621\]](#)

[Query transform output schema \[page 623\]](#)

[Output schema commands \[page 624\]](#)

When you right-click an element such as a column in the *Schema Out* pane of the Query transform, you can choose from several commands to control output content.

18.7.1 Schema In and Schema Out

The *Schema In* and *Schema Out* panes determine the fields that continue on in your data flow as well as field mapping.

The software helps you populate the schema panes in the Query transform based on objects and connections in your data flow. You can switch the focus in the *Query Editor* to change the current schema. You can also automatically complete the *Schema Out* pane with a target table schema.

There are several ways to change the current schema in a Query transform:

- Select a schema in the *Schema Out* pane.
- Right-click a schema, column, or function in the *Schema Out* pane and select *Make Current*.
- Double-click a schema in the *Schema Out* pane.

To automatically complete an empty output schema with a target table schema, connect the Query transform to a target table in your data flow. If the *Schema Out* pane is already populated, the software does not overwrite those mappings.

There are several techniques to map an input schema to an output schema:

- Drag and drop columns or nested schemas from the input schema area to the output schema area for simple column mappings. When you drop a single column on top of an existing column, you can remap that column.
 - Select *Remap Column* to update only the column mapping.
 - Select *Remap with Data Type* to update the column mapping a data type.
 - Select *Insert Above* or *Insert Below* to add a column as a new mapping.
- Right-click the current schema in the *Schema Out* pane and select *New Output Column* or *New Output Schema*. You can provide simple column mappings by dragging input columns over the new output columns. For complex mappings, use the options area.
- Right-click columns in the current schema in the *Schema Out* pane to assign and reverse primary key settings. A key icon indicates primary keys.
- Right-click the current schema in the *Schema Out* pane and select *New Function*.
 - Only functions that are in the repository are available.
 - Add adapter functions and SAP RFC or BAPI functions. These types of functions return multiple columns (in contrast to the functions used in mappings and Where clauses, which return single values).
 - Map all first-level input parameters for the function to the input parameters of the query.
- Right-click the current schema in the *Schema Out* pane and select *Unnest* to flatten nested output schemas. Use the *Unnest* option when the source has a nested schema, such as an XML file. You map columns from this source to a flat target table schema.
- Right-click elements in the *Schema Out* pane and select a command. The elements must be within the current schema.

Parent topic: [Query transform \[page 619\]](#)

Related Information

[Query transform input schema \[page 621\]](#)

[Query transform output schema \[page 623\]](#)

[Output schema commands \[page 624\]](#)

[Searching in an input or output schema \[page 626\]](#)

[Query transform output schema \[page 623\]](#)

[Output schema commands \[page 624\]](#)

18.7.2 Query transform input schema

The input schema area displays all schemas input to the Query transform as a hierarchical tree. Each input schema can contain zero or more of the following elements:

- Columns
- Nested schemas

Icons preceding columns are combinations of the following graphics:

Icon	Description
	Primary key.
	Column that is not used in output mapping.
	Column that is used in output mappings.

The *Input* list at the top left of the query editor indicates the schema that is currently selected. As you select schemas or columns in the input schema area, the *Input* list displays the corresponding schema. Conversely, you can select a schema in the *Input* list to move easily to a required schema.

You can right-click elements in the input schema area and select the following menu commands:

Command	Applicable elements	Effect
Copy	Columns, schemas	Stores a copy of the selected elements in the clipboard, leaving the elements in the input schema area.
Find	Anywhere in input schema area	Locates an output element with the name or description you enter.
Refresh	Anywhere in input schema area	Refreshes the display of the input schema area.
Parent	Columns	Selects the parent schema of the selected column.

Command	Applicable elements	Effect
Collapse	Columns, schemas	Collapses a selected schema or a selected column's parent schema (to facilitate viewing/navigation).
Generate DTD...		Generates a DTD format that corresponds to the structure of the selected schema (either NRDM or relational). Generates all data types as varchar.
Generate XML Schema...		Generates an XML Schema that corresponds to the structure of the selected schema (either NRDM or relational). All data types match those of the selected schema.
Propagate Column From...	Columns	Carries a selected column schema from an upstream source or transform through intermediate objects to the input schema. Simple mappings are created in each object with no change to the data type or data itself.
Map to Output	Columns, schemas	Creates a simple mapping from the input schema area to the output schema area.
Create File Format...		Creates a file format from a relational table schema. All data types match those of the original table schema.
Properties	Columns, schemas	Displays the properties of the selected element. You cannot modify the properties.

Parent topic: [Query transform \[page 619\]](#)

Related Information

[Schema In and Schema Out \[page 620\]](#)

[Query transform output schema \[page 623\]](#)

[Output schema commands \[page 624\]](#)

[Searching in an input or output schema \[page 626\]](#)

18.7.3 Query transform output schema

The output schema area displays the schema output from the Query transform as a hierarchical tree. The output schema can contain one or more of the following elements:

- Columns
- Nested schemas
- Functions

Icons preceding columns are combinations of the following graphics:

Icon	Description
	Primary key.
	Column that has a simple mapping. A simple mapping is either a single column or an expression with no input column (that is, an expression that does not vary with input).
	Column that has a complex mapping. A complex mapping is any mapping that is not simple.
	(Red cross superimposed on any icon) Incorrect mapping. Data Services does not perform a complete validation during design, so the editor may not flag an incorrect mapping. For a complete validation, select Validation  Validate  .

The Schema Out pane shows the following:

- The current schema in the Schema Out list at the top and in the output schema area. The current schema determines:
 - The output elements that you can modify (add, map, or delete).
 - The scope of the SELECT through ORDER BY tabs in the options area.
- Non-current schemas appear dim in the output schema area.

Parent topic: [Query transform \[page 619\]](#)

Related Information

[Schema In and Schema Out \[page 620\]](#)

[Query transform input schema \[page 621\]](#)

[Output schema commands \[page 624\]](#)

[Searching in an input or output schema \[page 626\]](#)

[Schema In and Schema Out \[page 620\]](#)

18.7.4 Output schema commands

When you right-click an element such as a column in the *Schema Out* pane of the Query transform, you can choose from several commands to control output content.

The following table contains descriptions for the commands available to select.

Command	Applicable elements	Effect
<i>Cut</i>	All	Removes the selected elements from the <i>Schema Out</i> pane and stores a copy of the elements in the clipboard.
<i>Copy</i>	All	Stores a copy of the selected elements in the clipboard, leaving the elements in the <i>Schema Out</i> pane.
<i>Paste</i>	All	Inserts the elements stored in the clipboard at the current cursor location (this must be within the current schema). Only visible when the clipboard contains something. If the cursor overlaps an existing column, you are prompted to insert above, insert below, remap column, or cancel.
i Note Copying an input element and pasting it in the <i>Schema Out</i> pane provides a simple mapping from the input element to the output element. You can also do this by dragging the input element to the <i>Schema Out</i> pane.		
<i>Delete</i>	All	Removes the selected elements from the <i>Schema Out</i> pane, without making a copy.
<i>Find...</i>	All	Locates an output element with the name or description you enter.
<i>Make Current</i>	All except for the current schema	Makes the selected schema, or the schema of the selected element, the current schema.
<i>New Output Column...</i>	Schemas	Adds an output column to the current schema with the name and properties you enter.
<i>New Output Schema...</i>	Schemas	Adds a nested schema to the current schema with the name you enter.

Command	Applicable elements	Effect
<i>New Function Call</i>	Schemas	Adds a function or stored procedure call to the current schema. The function or procedure must already be imported into the repository. You can add adapter functions and SAP RFC or BAPI functions. These functions return multiple columns (in contrast to the functions used in mappings and Where clauses that return single values). In the <i>Define Input Parameter(s)</i> window, map all first-level input parameters for the function to the input parameters of the query.
<i>Modify Function Call</i>	Functions	Allows you to modify the selected function. In the <i>Define Input Parameter(s)</i> window, map all first-level input parameters for the function to the input parameters of the query.
<i>Propagate Column From</i>	Columns	Carries a selected column schema from an upstream source or transform through intermediate objects to the output schema. Simple mappings are created in each object with no change to the data type or data itself.
<i>Unnest</i>	Nested schemas	Toggles to flatten or re-nest the selected schema into the parent schema in the query output. An unnested schema will not appear in the succeeding transform or target; only its columns appear. Unnested schemas appear in the Query transform <i>Schema Out</i> pane as table icons with a black arrow that points to the left.
<i>Nest with sub-schemas</i>	Nested schemas	Re-nests the selected schema and all sub-schemas into the parent schema in the query output.
<i>Unnest with sub-schemas</i>	Nested schemas	Flattens the selected schema and all sub-schemas into the parent schema in the query output. Unnested schemas and sub-schemas do not appear in the succeeding transform or target; only the columns appear. Unnested schemas and sub-schemas appear in the Query transform <i>Schema Out</i> pane as table icons with a black arrow that points to the left.
<i>Primary Key</i>	Columns	Toggles the primary key attribute of the column on (check mark appears next to the command) or off (no check mark appears next to the command). A key icon indicates that a column is a primary key.
<i>Optional</i>	Schemas	Toggles to make a schema optional.
<i>Generate DTD</i>	All	Generates a DTD format that corresponds to the structure of the selected schema (either NRDM or relational). Generates all data types as varchar.

Command	Applicable elements	Effect
Generate XML Schema	Schema	Generates an XML Schema that corresponds to the structure of the selected schema (either NRDM or relational). All data types match those of the selected schema.
Create File Format	Schema	Creates a file format from a relational table schema. All data types match those of the original table schema.
Properties	All	Displays the properties of the selected element.

Parent topic: [Query transform \[page 619\]](#)

Related Information

[Schema In and Schema Out \[page 620\]](#)

[Query transform input schema \[page 621\]](#)

[Query transform output schema \[page 623\]](#)

[Searching in an input or output schema \[page 626\]](#)

18.7.5 Searching in an input or output schema

1. In the Query editor Find tab, enter the search term in the [Find what](#) box or select from previous search terms in the drop-down list.
2. In the [Schemas](#) list, choose the schemas in which to search.
3. In the [Elements](#) list, choose the types of mappings in which to search.
4. In the [Where](#) list, choose the properties to search within.

i Note

You can search within one or all properties, but not within two or three specific properties at a time.

5. Select the [Match case](#) check box to constrain your search to the capitalization entered.
6. Click [Find](#).
The Designer searches the query for the words you specified within the constraints you defined.

i Note

The Designer searches for columns loaded into memory. If columns are not loaded into memory, you must expand the schema to load the columns into memory before clicking [Find](#) and searching for the columns.

All matches are shown in the box below the find constraints. When you click to select a table or column name, the table or column is automatically highlighted and shown in the corresponding input or output schema area.

Initially, the Designer lists the matching columns in the order that they appear within the schemas. If you are searching both schemas, the Designer lists the first match found in the input schema first and the last match found in the output schema last. You can sort the list of matches by property. Each time you click a property heading, the Designer resorts the matches, cycling through original order, ascending order, and descending order.

Arrow icons confirm column and sort type. For example, if you sort the data by the [Description](#) property and in ascending order, an "up" arrow appears next to the [Description](#) heading. Click the heading again and a "down" arrow appears to indicate that the data is now sorted in descending alpha-numeric order. Click again and the match list returns to its original sort order.

Task overview: [Query transform \[page 619\]](#)

Related Information

[Schema In and Schema Out \[page 620\]](#)

[Query transform input schema \[page 621\]](#)

[Query transform output schema \[page 623\]](#)

[Output schema commands \[page 624\]](#)

18.8 Entity extraction

The extraction process uses inherent knowledge of the semantics of words and the linguistic context in which these words occur to find entities and facts.

The extraction process creates specific patterns to extract entities and facts based on system rules. You can add entries to a dictionary and you can write custom rules to customize extraction output.

The following example shows how unstructured content is transformed into structured information for further processing and analysis.

Example: Sample text and extraction information

Example of your data: "Mr. Jones is very upset with Green Insurance Corp. The offer for his totaled vehicle is too low. He states that Green offered him \$1250.00 but his car is worth anywhere from \$2500 to \$4500. Mr. Jones would like Green's comprehensive coverage to be in line with other competitors."

The Entity Extraction transform processes the text by configuring the sentiment with custom rules. It identifies the group in a logical way to identify the various parts of the entity and by identifying the facts.

The following tables show partial results with information tagged as entities, types, and subtypes.

Entity	Type	Subtype
Mr. Jones	Person	N/A
Green Insurance	Organization	Commercial
Green	Prop_Misc	N/A
\$1250.00	currency	

The following table shows partial results with information tagged as facts, types, subfacts, and subfact types.

Fact	Type	Subfact	Subfact type
Mr. Jones is very upset with Green Insurance Corp.	Sentiment	Very upset	StrongNegativeSentiment
Jones would like that Green's comprehensive coverage to be in line with other competitors.	Request	N/A	N/A

18.8.1 Use the Entity Extraction transform

Use the Entity Extraction transform to extract information from any text, HTML, XML, or binary-formats such as PDF, and to generate structured output.

Use the generated output in several ways based on your work flow. Use it as an input to another transform or write to multiple output sources such as a database table or a flat file. The transform generates the output in UTF-16 encoding. The following list provides examples to show when you would use the Entity Extraction transform:

- To search for specific information and relationships from a large amount of text related to a broad domain. For example, a company is interested in analyzing customer feedback received in free-form text after a new product launch.
- To link structured information from unstructured text together with existing structured information to make new connections. For example, a law enforcement department is trying to make connections between various crimes and people involved using their own database and information available in various reports in text format.
- To analyze and report on product quality issues such as excessive repairs and returns for certain products by making connections between various forms of data. For example, you could have product information in structured data format such as products, parts, customers, and suppliers. Still other data contains text about product issues and customer complaints. A third data source contains maintenance records, repair logs, product escalations, and support center logs.

18.8.2 Entities, types, subtypes, and facts

The Entity Extraction transform pulls information from your data and creates entity types and subtypes. It further categorizes data into different types of facts.

Entities

In Data Services, entities are names of people, places, and things that the Entity Extraction transform extracts from your data.

❖ Example

Entities

Entity	Type
Paris	Locality
Mr. Joe Smith	Person

Subtypes

Entities can have subtypes, which are subcategories of an entity. Subtypes indicates further classification of an entity. Subtypes are hierarchical specifications of an entity type that enables the distinction between different semantic varieties of the same entity type.

❖ Example

Types and subtypes

Entity	Type	Subtype
Boeing 747	Vehicle	Air
Mercedes-Benz SL500	Vehicle	Land
SAP	Organization	Commercial

Facts

Entities by themselves only show that they are present in a document. Facts provide information about how entities are related. The term “Fact” covers multiple meanings: Complex patterns that include one or more entities, relationships between one or more entities, or predicates about an entity.

Facts indicate a pattern. The transform turns the pattern into extraction rules that extract such information as sentiments, events, or relationships. The facts provide the context about how different entities are connected in the text. The Entity Extraction transform also uses custom extraction rules that you create.

Just like entities, facts have types and subtypes.

- Fact types identify the category of a fact, such as sentiments and requests.
- Fact subtypes or subfacts are key pieces of information embedded within a fact.
- Subfact types are like categories associated with the subfact

❖ Example

Examples of Facts and fact types

Fact	Type
SAP acquired Business Objects in a friendly takeover.	Merger and acquisitions
Mr. Joe Smith is upset with is airline bookings.	Sentiment

18.8.3 Differences between text data processing and data cleanse transforms

The Entity Extraction transform provides functionality similar to the Data Cleanse transform in certain cases, especially with respect to customization capabilities.

The documentation describes the differences between the two and which transform to use to meet your goals. The Text Data Processing Entity Extraction transform is for making sense of unstructured content and the Data Cleanse transform is for standardizing and cleansing structured data. The following table describes some of the main differences. In many cases, using a combination of Entity Extraction and Data Cleanse transforms will generate the data that is best suited for your business intelligence analyses and reports.

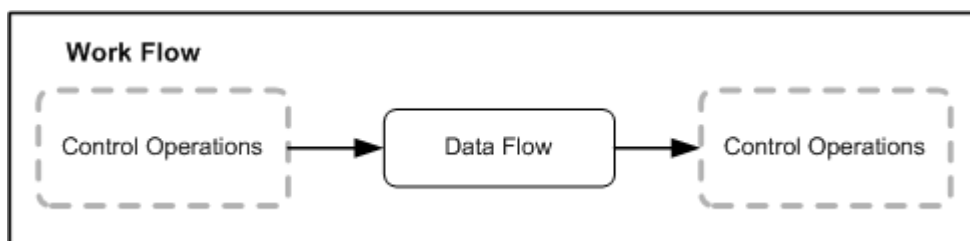
Criteria	Text Data Processing	Data Cleanse
Input type	Unstructured text that requires linguistic parsing to generate relevant information.	Structured data represented as fields in records.
Input size	More than 5KB of text.	Less than 5KB of text.
Input scope	Normally broad domain with many variations.	Specific data domain with limited variations.

Criteria	Text Data Processing	Data Cleanse
Matching task	Content discovery, noise reduction, pattern matching, and relationship between different entities.	Dictionary lookup, pattern matching.
Potential usage	Identifies potentially meaningful information from unstructured content and extracts it into a format that can be stored in a repository.	Ensures quality of data for matching and storing into a repository such as Meta Data Management.
Output	Creates annotations about the source text in the form of entities, entity types, facts, and their offset, length, and so on. Input is not altered.	Creates parsed and standardized fields. Input is altered if desired.

19 Work flows

A work flow defines the decision-making process for executing data flows.

For example, elements in a work flow can determine the path of execution based on a value set by a previous job or can indicate an alternative path if something goes wrong in the primary path. Ultimately, the purpose of a work flow is to prepare for executing data flows and to set the state of the system after the data flows are complete.



Jobs are special work flows. Jobs are special because you can execute them. Almost all of the features documented for work flows also apply to jobs, with one exception: jobs do not have parameters.

Related Information

[Projects \[page 86\]](#)

19.1 Steps in a work flow

Work flow steps take the form of icons that you place in the work space to create a work flow diagram.

The following objects can be elements in work flows:

- Work flows
- Data flows
- Scripts
- Conditionals
- While loops
- Try/catch blocks

Work flows can call other work flows, and you can nest calls to any depth. A work flow can also call itself.

The connections you make between the icons in the workspace determine the order in which work flows execute, unless the jobs containing those work flows execute in parallel.

19.2 Order of execution in work flows

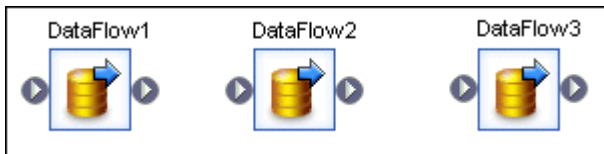
Steps in a work flow execute in a left-to-right sequence indicated by the lines connecting the steps.

Here is the diagram for a work flow that calls three data flows:

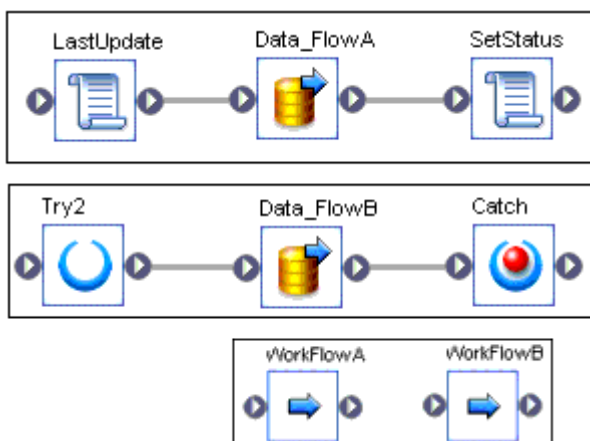


Note that Data_Flow1 has no connection from the left but is connected on the right to the left edge of Data_Flow2 and that Data_Flow2 is connected to Data_Flow3. There is a single thread of control connecting all three steps. Execution begins with Data_Flow1 and continues through the three data flows.

Connect steps in a work flow when there is a dependency between the steps. If there is no dependency, the steps need not be connected. In that case, the software can execute the independent steps in the work flow as separate processes. In the following work flow, the software executes data flows 1 through 3 in parallel:



To execute more complex work flows in parallel, define each sequence as a separate work flow, then call each of the work flows from another work flow as in the following example:



You can specify that a job execute a particular work flow or data flow only one time. In that case, the software only executes the first occurrence of the work flow or data flow; the software skips subsequent occurrences in the job. You might use this feature when developing complex jobs with multiple paths, such as jobs with try/

catch blocks or conditionals, and you want to ensure that the software only executes a particular work flow or data flow one time.

19.3 Example of a work flow

Suppose you want to update a fact table. You define a data flow in which the actual data transformation takes place. However, before you move data from the source, you want to determine when the fact table was last updated so that you only extract rows that have been added or changed since that date.

You need to write a script to determine when the last update was made. You can then pass this date to the data flow as a parameter.

In addition, you want to check that the data connections required to build the fact table are active when data is read from them. To do this in the software, you define a try/catch block. If the connections are not active, the catch runs a script you wrote, which automatically sends mail notifying an administrator of the problem.

Scripts and error detection cannot execute in the data flow. Rather, they are steps of a decision-making process that influences the data flow. This decision-making process is defined as a work flow, which looks like the following:



The software executes these steps in the order that you connect them.

19.4 Creating work flows

You can create work flows through the Object Library or the Tool Palette.

After creating a work flow, you can specify that a job only execute the work flow one time, as a single process, or as a continuous process even if the work flow appears in the job multiple times.

19.4.1 Creating a new work flow using the object library

1. Open the object library.
2. Go to the [Work Flows](#) tab.
3. Right-click and choose [New](#).
4. Drag the work flow into the diagram.
5. Add the data flows, work flows, conditionals, try/catch blocks, and scripts that you need.

19.4.2 Creating a new work flow using the tool palette

1. Select the work flow icon in the tool palette.
2. Click where you want to place the work flow in the diagram.

If more than one instance of a work flow appears in a job, you can improve execution performance by running the work flow only one time.

19.4.3 Specifying that a job executes the work flow one time

When you specify that a work flow should only execute once, a job will never re-execute that work flow after the work flow completes successfully.

The exception is if the work flow is contained in a work flow that is a recovery unit that re-executes and has not completed successfully elsewhere outside the recovery unit.

It is recommended that you not mark a work flow as *Execute only once* if the work flow or a parent work flow is a recovery unit.

1. Right click on the work flow and select *Properties*.
The Properties window opens for the work flow.
2. Select *Regular* from the *Execution type* dropdown list.
3. Select the *Execute only once* check box.
4. Click *OK*.

Related Information

[Reference Guide: Work flow](#)

19.4.4 What is a single work flow?

A single work flow runs all of its child data flows in one operating system process.

If the data flows are designed to be run in parallel then they are run in different threads instead of different processes. The advantage of single process is that it is possible to share resources such as database connections across multiple data flows.

i Note

Single work flows have the following limitations:

- A single work flow cannot call or contain a continuous work flow.
- A single work flow cannot use sub data flows. Therefore, the Data Transfer transform and "Run as a separate process" options are invalid. The software will generate a runtime validation error.

- A single work flow can be only executed by a continuous work flow. A single work flow cannot call another single work flow.

19.4.4.1 Specifying that a job executes as a single work flow

1. Right-click a work flow and select *Properties*
2. Select *Single* from the Execution type dropdown list.
3. Click *OK*.

19.4.5 What is a continuous work flow?

A continuous work flow runs all data flows in a loop but keeps them in the memory for the next iteration.

This eliminates the need to repeat some of the common steps of execution for example, connecting to the repository, parsing—optimizing—compiling ATL, and opening database connections.

i Note

Continuous work flows have the following limitations:

- A continuous work flow cannot call or contain another continuous work flow. If a continuous work flow calls another continuous work flow, which never terminates, the work flow can never restart the child processes.
- A continuous work flow cannot use sub data flows. Therefore, the Data Transfer transform and "Run as a separate process" options are invalid. The software will generate a runtime validation error.
- A regular or single work flow cannot call or contain a continuous work flow.
- A real-time job cannot call a continuous work flow.
- The platform transform XML_Map cannot be used in a continuous work flow.
- The Data Integrator transforms Data_Transfer and XML_Pipeline cannot be used in a continuous work flow.
- Transaction control in a continuous work flow is supported for Oracle databases only.

19.4.5.1 Specifying that a job executes a continuous work flow

1. Right-click a work flow and select *Properties*
2. Select *Continuous* from the Execution type dropdown list.
3. Access the *Continuous Options* tab.
4. Specify when you want the work flow to release resources:
 - To release resources after a number of runs, select *Number of runs* and enter the number of runs. The default is 100.

- To release resources after a number of hours, select the *After* checkbox, select *Number of hours*, and enter the number of hours.
 - To release resources after a number of days, select the *After* checkbox, select *Number of days*, and enter the number of days.
 - To release resources when the result of a function is not equal to zero, select the *After* checkbox, select *Result of the function is not equal to zero*, and enter the function you want to use.
5. To stop the work flow when the result of a custom function is equal to zero, select *When result of the function is equal to zero*, and enter the custom function you want to use.
6. Click *OK*.

19.5 Conditionals

Conditionals are single-use objects used to implement if/then/else logic in a work flow.

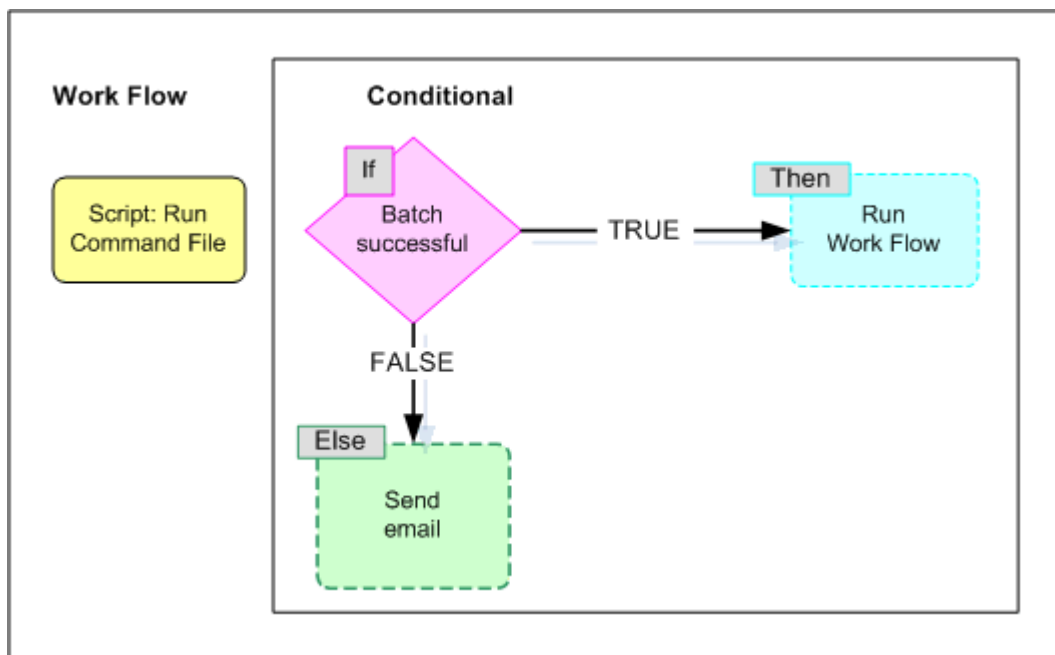
Conditionals and their components (if expressions, then and else diagrams) are included in the scope of the parent control flow's variables and parameters.

To define a conditional, you specify a condition and two logical branches:

Conditional branch	Description
<i>If</i>	A Boolean expression that evaluates to TRUE or FALSE. You can use functions, variables, and standard operators to construct the expression.
<i>Then</i>	Work flow elements to execute if the <i>If</i> expression evaluates to TRUE.
<i>Else</i>	(Optional) Work flow elements to execute if the <i>If</i> expression evaluates to FALSE.

Define the *Then* and *Else* branches inside the definition of the conditional.

A conditional can fit in a work flow. Suppose you use a Windows command file to transfer data from a legacy system into the software. You write a script in a work flow to run the command file and return a success flag. You then define a conditional that reads the success flag to determine if the data is available for the rest of the work flow.



To implement this conditional in the software, you define two work flows—one for each branch of the conditional. If the elements in each branch are simple, you can define them in the conditional editor itself.

Both the [Then](#) and [Else](#) branches of the conditional can contain any object that you can have in a work flow including other work flows, nested conditionals, try/catch blocks, and so on.

19.5.1 Defining a conditional

1. Define the work flows that are called by the [Then](#) and [Else](#) branches of the conditional.

It is recommended that you define, test, and save each work flow as a separate object rather than constructing these work flows inside the conditional editor.

2. Open the work flow in which you want to place the conditional.
3. Click the icon for a conditional in the tool palette.
4. Click the location where you want to place the conditional in the diagram.

The conditional appears in the diagram.

5. Click the name of the conditional to open the conditional editor.
6. Click [if](#).
7. Enter the Boolean expression that controls the conditional.

Continue building your expression. You might want to use the function wizard or smart editor.

8. After you complete the expression, click [OK](#).
9. Add your predefined work flow to the [Then](#) box.

To add an existing work flow, open the object library to the Work Flows tab, select the desired work flow, then drag it into the [Then](#) box.

10. (Optional) Add your predefined work flow to the [Else](#) box.

If the If expression evaluates to FALSE and the Else box is blank, the software exits the conditional and continues with the work flow.

11. After you complete the conditional, choose [DebugValidate](#).

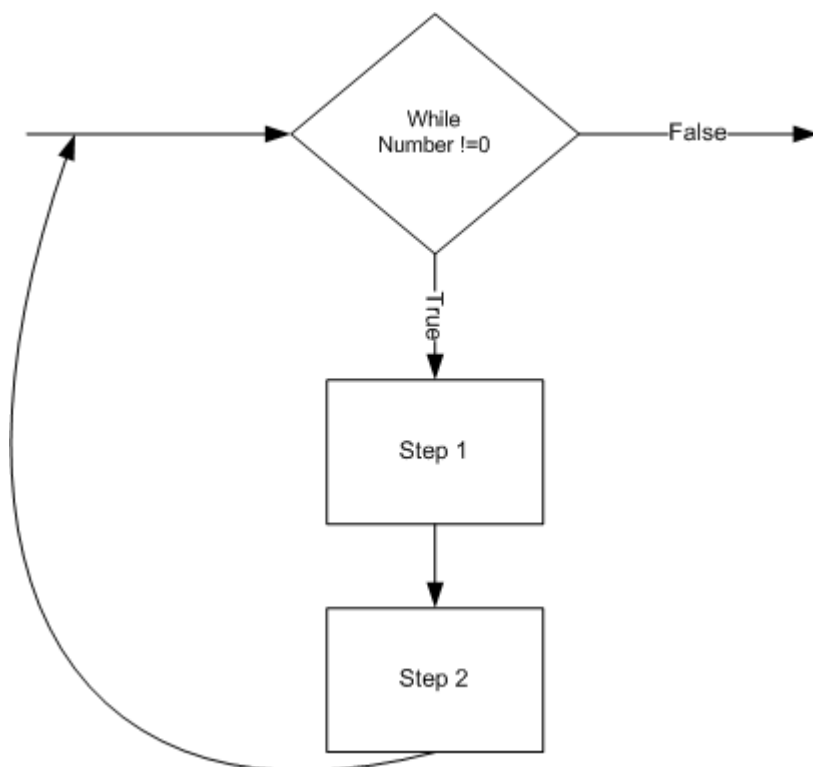
The software tests your conditional for syntax errors and displays any errors encountered.

12. The conditional is now defined. Click the [Back](#) button to return to the work flow that calls the conditional.

19.6 While loops

The while loop is a single-use object that you can use in a work flow.

Use a while loop to repeat a sequence of steps in a work flow as long as a condition is true.

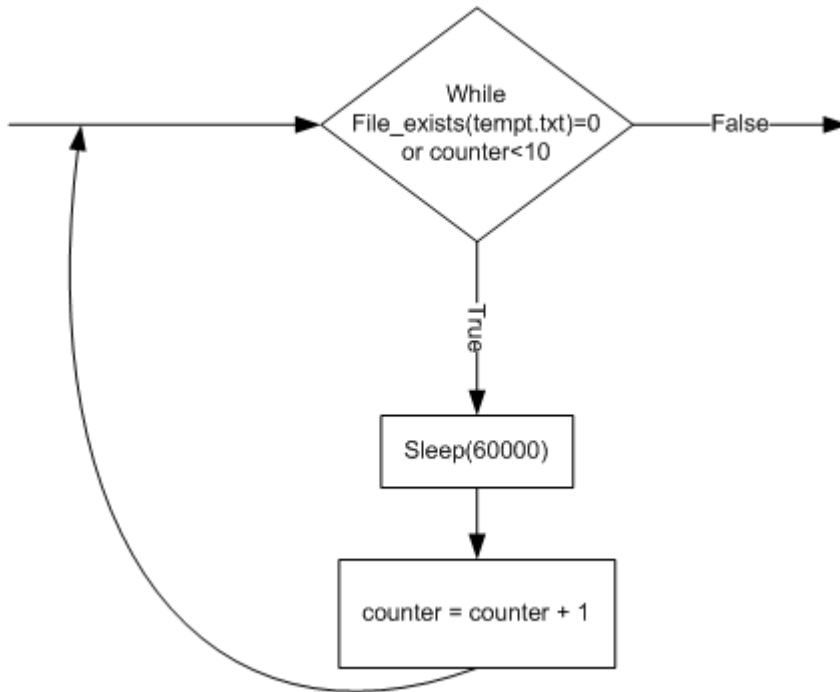


Typically, the steps done during the while loop result in a change in the condition so that the condition is eventually no longer satisfied and the work flow exits from the while loop. If the condition does not change, the while loop will not end.

For example, you might want a work flow to wait until the system writes a particular file. You can use a while loop to check for the existence of the file using the `file_exists` function. As long as the file does not exist, you can have the work flow go into sleep mode for a particular length of time, say one minute, before checking again.

Because the system might never write the file, you must add another check to the loop, such as a counter, to ensure that the while loop eventually exits. In other words, change the while loop to check for the existence of

the file and the value of the counter. As long as the file does not exist and the counter is less than a particular value, repeat the while loop. In each iteration of the loop, put the work flow in sleep mode and then increment the counter.



19.6.1 Defining a while loop

You can define a while loop in any work flow.

1. Open the work flow where you want to place the while loop.
2. Click the while loop icon on the tool palette.
3. Click the location where you want to place the while loop in the workspace diagram.

The while loop appears in the diagram.

4. Click the while loop to open the while loop editor.
5. In the *While* box at the top of the editor, enter the condition that must apply to initiate and repeat the steps in the while loop.

Alternatively, you can open the expression editor, which gives you more space to enter an expression and access to the function wizard. Click *OK* after you enter an expression in the editor.

6. Add the steps you want completed during the while loop to the workspace in the while loop editor.

You can add any objects valid in a work flow including scripts, work flows, and data flows. Connect these objects to represent the order that you want the steps completed.

i Note

Although you can include the parent work flow in the while loop, recursive calls can create an infinite loop.

7. After defining the steps in the while loop, choose  **Debug**  **Validate** .

The software tests your definition for syntax errors and displays any errors encountered.

8. Close the while loop editor to return to the calling work flow.

19.6.2 Using a while loop with View Data

Depending on the design of your job, the software might not complete all iterations of a while loop if you run a job in view data mode.

When using View Data, a job stops when the software has retrieved the specified number of rows for all scannable objects.

The following might occur when using while loop when running a job in view data mode:

- If the while loop contains scannable objects and there are no scannable objects outside the while loop (for example, if the while loop is the last object in a job), then the job will complete after the scannable objects in the while loop are satisfied, possibly after the first iteration of the while loop.
- If there are scannable objects after the while loop, the while loop will complete normally. Scanned objects in the while loop will show results from the last iteration.
- If there are no scannable objects following the while loop but there are scannable objects completed in parallel to the while loop, the job will complete as soon as all scannable objects are satisfied. The while loop might complete any number of iterations.

19.7 Try/catch blocks

A try/catch block is a combination of one try object and one or more catch objects that allow you to specify alternative work flows if errors occur while the software is executing a job.

Try/catch blocks:

- "Catch" groups of exceptions "thrown" by the software, the DBMS, or the operating system.
- Apply solutions that you provide for the exceptions groups or for specific errors within a group.
- Continue execution.

Try and catch objects are single-use objects.

Here's the general method to implement exception handling:

1. Insert a try object before the steps for which you are handling errors.
2. Insert a catch object in the work flow after the steps.
3. In the catch object, do the following:
 - Select one or more groups of errors that you want to catch.
 - Define the actions that a thrown exception executes. The actions can be a single script object, a data flow, a work flow, or a combination of these objects.
 - Optional. Use catch functions inside the catch block to identify details of the error.

If an exception is thrown during the execution of a try/catch block and if no catch object is looking for that exception, then the exception is handled by normal error logic.

The following work flow shows a try/catch block surrounding a data flow:



In this case, if the data flow BuildTable causes any system-generated exceptions specified in the catch Catch_A, then the actions defined in Catch_A execute.

The action initiated by the catch object can be simple or complex. Here are some examples of possible exception actions:

- Send the error message to an online reporting database or to your support group.
- Rerun a failed work flow or data flow.
- Run a scaled-down version of a failed work flow or data flow.

Related Information

[Defining a try/catch block \[page 642\]](#)

[Categories of available exceptions \[page 643\]](#)

[Example: Catching details of an error \[page 644\]](#)

[Reference Guide: Objects, Catch](#)

19.7.1 Defining a try/catch block

To define a try/catch block:

1. Open the work flow that will include the try/catch block.
2. Click the try icon in the tool palette.
3. Click the location where you want to place the try in the diagram.

The try icon appears in the diagram.

i Note

There is no editor for a try; the try merely initiates the try/catch block.

4. Click the catch icon in the tool palette.
5. Click the location where you want to place the catch object in the work space.

The catch object appears in the work space.

6. Connect the try and catch objects to the objects they enclose.
7. Click the name of the catch object to open the catch editor.
8. Select one or more groups from the list of [Exceptions](#).

To select all exception groups, click the check box at the top.

9. Define the actions to take for each exception group and add the actions to the catch work flow box. The actions can be an individual script, a data flow, a work flow, or any combination of these objects.
 - a. It is recommended that you define, test, and save the actions as a separate object rather than constructing them inside the catch editor.
 - b. If you want to define actions for specific errors, use the following catch functions in a script that the work flow executes:
 - `error_context()`
 - `error_message()`
 - `error_number()`
 - `error_timestamp()`
 - c. To add an existing work flow to the catch work flow box, open the object library to the Work Flows tab, select the desired work flow, and drag it into the box.
10. After you have completed the catch, choose **Validation > Validate > All Objects in View**.

The software tests your definition for syntax errors and displays any errors encountered.
11. Click the [Back](#) button to return to the work flow that calls the catch.
12. If you want to catch multiple exception groups and assign different actions to each exception group, repeat steps 4 through 11 for each catch in the work flow.

i Note

In a sequence of catch blocks, if one catch block catches an exception, the subsequent catch blocks will not be executed. For example, if your work flow has the following sequence and Catch1 catches an exception, then Catch2 and CatchAll will not execute.

```
Try > DataFlow1 > Catch1 > Catch2 > CatchAll
```

If any error in the exception group listed in the catch occurs during the execution of this try/catch block, the software executes the catch work flow.

Related Information

[Categories of available exceptions \[page 643\]](#)

[Example: Catching details of an error \[page 644\]](#)

[Reference Guide: Objects, Catch](#)

19.7.2 Categories of available exceptions

Categories of available exceptions include:

- Execution errors (1001)
- Database access errors (1002)
- Database connection errors (1003)
- Flat file processing errors (1004)

- File access errors (1005)
- Repository access errors (1006)
- SAP system errors (1007)
- System resource exception (1008)
- SAP BW execution errors (1009)
- XML processing errors (1010)
- COBOL copybook errors (1011)
- Excel book errors (1012)
- Data Quality transform errors (1013)

19.7.3 Example: Catching details of an error

This example illustrates how to use the error functions in a catch script. Suppose you want to catch database access errors and send the error details to your support group.

1. In the catch editor, select the exception group that you want to catch. In this example, select the checkbox in front of [Database access errors \(1002\)](#).
2. In the work flow area of the catch editor, create a script object with the following script:

```
mail_to('support@my.com',
  'Data Service error number' || error_number(),
  'Error message: ' || error_message(),20,20);
print('DBMS Error: ' || error_message());
```

3. This sample catch script includes the mail_to function to do the following:
 - Specify the email address of your support group.
 - Send the error number that the error_number() function returns for the exception caught.
 - Send the error message that the error_message() function returns for the exception caught.
4. The sample catch script includes a print command to print the error message for the database error.

Related Information

[Reference Guide: Objects, Description of objects, Catch, Catch error functions](#)

[Reference Guide: Objects, Description of objects, Catch scripts](#)

19.7.4 Catch best practices

Use these best practice suggestions to set up a successful try/catch block in your job setup.

For each catch object in the try/catch block, specify the following:

- One or more groups of exceptions that the catch object handles.

i Note

If you want to assign different actions to different exception groups, add a catch for each set of actions.

- The actions to execute when an exception in the indicated exception groups occurs.
Optional but recommended: Define, test, and save the actions as a separate object rather than constructing them inside the catch editor. The actions can be a single script object, a data flow, a work flow, or a combination of these objects.
- Optional error functions inside the catch block to identify details of the error.

If an exception is thrown during the execution of a try/catch block, and if no catch object is looking for that exception group, then the block handles the exception with normal error logic.

For batch jobs only, do not reference output variables from a try/catch block in any subsequent steps if you are using the automatic recovery feature. Referencing such variables could alter the results during automatic recovery.

You can use try/catch blocks in any real-time job component. However, try/catch blocks cannot straddle a real-time processing loop and the initialization or cleanup component of a real-time job.

19.8 Scripts

Scripts are single-use objects used to call functions and assign values to variables in a work flow.

For example, you can use the SQL function in a script to determine the most recent update time for a table and then assign that value to a variable. You can then assign the variable to a parameter that passes into a data flow and identifies the rows to extract from a source.

A script can contain the following statements:

- Function calls
- If statements
- While statements
- Assignment statements
- Operators

The basic rules for the syntax of the script are as follows:

- Each line ends with a semicolon (;).
- Variable names start with a dollar sign (\$).
- String values are enclosed in single quotation marks (').
- Comments start with a pound sign (#).
- Function calls always specify parameters even if the function uses no parameters.

For example, the following script statement determines today's date and assigns the value to the variable \$TODAY:

```
$TODAY = sysdate();
```

You cannot use variables unless you declare them in the work flow that calls the script.

Related Information

[Reference Guide: Scripting Language \[page 803\]](#)

19.8.1 Creating a script

1. Open the work flow.
2. Click the script icon in the tool palette.
3. Click the location where you want to place the script in the diagram.

The script icon appears in the diagram.

4. Click the name of the script to open the script editor.
5. Enter the script statements, each followed by a semicolon.

The following example shows a script that determines the start time from the output of a custom function.

```
AW_StartJob ('NORMAL','DELTA', $G_STIME,$GETIME);
$GETIME =to_date(
sql('ODS_DS','SELECT to_char(MAX(LAST_UPDATE) ,
\'YYYY-MM-DD HH24:MI:SS\')
FROM EMPLOYEE'),
'YYYY_MMM_DDD_HH24:MI:SS');
```

Click the function button to include functions in your script.

6. After you complete the script, select  **Validation**  **Validate** .

The software tests your script for syntax errors and displays any errors encountered.

7. Click the ... button and then **save** to name and save your script.

The script is saved by default in [<LINK_DIR>](#)/Data Services/ DataQuality/Samples.

19.8.2 Debugging scripts using the print function

The software has a debugging feature that allows you to print:

- The values of variables and parameters during execution
- The execution path followed within a script

You can use the print function to write the values of parameters and variables in a work flow to the trace log. For example, this line in a script:

```
print('The value of parameter $<x>: [$<x>]');
```

produces the following output in the trace log:

```
The following output is being printed via the Print function in <Session
<job_name>.
The value of parameter $<x>: <value>
```

Related Information

[Reference Guide: Functions and Procedures, Descriptions of built-in functions, print](#)

19.9 Smart Editor and function tools

SAP Data Services provides the Smart Editor and other function tools to help you create scripts, expressions, and custom functions.

With Smart Editor, there is no need to type the names of existing elements like column, function, and variable names. The software provides access to the Smart Editor and function tools right where you need them. For example, open both tools in the Query transform editor when you create a WHERE clause. Also open the Smart Editor from the [Functions](#) tab in the object library when you create or edit a custom function.

i Note

Sometimes we refer to the function tools as the Function wizard.

While you use the Smart Editor to create functions, open the [Select Function](#) dialog box to develop custom functions. The [Select Function](#) dialog box lists several function categories in the left pane. Select a category and the function names appear in the right pane. Select a function name and click [Next](#) to continue defining the function.

Related Information

[Scripting Language \[page 803\]](#)

[Custom functions \[page 778\]](#)

[Lookup tables and the lookup_ext function \[page 288\]](#)

19.9.1 Accessing the Smart Editor

SAP Data Services places access to the Smart Editor in places where you most need it.

In the following steps we use an example scenario to help you understand one way to open the Smart Editor from Data Services.

In a data flow that includes a Query transform:

1. Open the Query editor.
2. Right-click in a blank area of the [Mapping](#) tab. Select [Enable ToolTips](#) and [Enable Selection List](#) from the dropdown menu.
3. Click away from the dropdown menu to close it.

Perform the following steps to open the Smart Editor and function wizard from the Query editor.

1. Drag a column from the *Schema In* pane of the Query editor to the *Schema Out* pane.

The software adds the column mapping to the *Mapping* tab, and enables the Smart Editor and function tools icons.

2. Add to the mapping expression by entering text.

The Smart Editor opens the Selection List showing suggestions for the text you started to enter.

3. Select the suggested function and continue creating the expression.
4. Map another field by dragging it from the Schema In pane to the Schema Out pane.
5. Open the Smart Editor by clicking the ellipses icon. Open the function wizard by clicking *Functions...*

When you open the smart editor window, the software displays the context of the object from which you opened it in the title bar.

You cannot add comments to a mapping clause in a Query transform. The job does not run and you cannot successfully export it. Use the object description or workspace annotation feature instead.

Open the smart editor from the following locations:

- Query Editor *Mapping* tab
- Query Editor *From* tab
- Query Editor *Where* tab
- Script Editor
- Conditional Editor
- While Loop Editor
- Custom Function Editor
- Function wizard, "Define Input Parameter(s)" page
- SQL - Transform Editor
- Case Editor

19.9.2 Using the selection list and a tool tips

The Smart Editor enables experienced users to create expressions without the aid of the library pane by enabling two tools from the menu list: Selection list and tool tips.

Open the Smart Editor from a Query transform, or other applicable object in your data flow.

1. Right-click in a blank area of the editor pane.

A dropdown menu appears.

2. Select the *Enable Selection List* and *Enable Tool Tip* items so that a checkmark appears next to them.
3. Click off of the menu to close the menu.
4. Optional. Click the *Show/Hide Editor Library* icon in the tool bar to close the library pane at left.
5. Begin to type an element of your expression or custom function in the editor pane.

A list of functions appears. For example, start to type "avg", the function list appears, select **avg** from the function list. The software populates the editor pane with **avg (**.

6. Click at the end of the function to view a tool tip that contains the correct syntax.

For example, the tool tip for **avg (** is **avg ([in] ColumnName) .** The tool tip contains the same description, definition, or syntax that you see if you selected it from the library pane. The tool tip remains

on the screen so that you can follow the syntax of the function. If you enter an input value of the wrong data type, the tool tip closes, indicating an error.

19.9.3 Browsing for a function

Use the browsing tools in the library pane of the Smart Editor to browse for a function.

1. Open the *Functions* tab in the library pane and expand the applicable function group.

The Smart Editor groups the function categories in the following ways:

- Groups built-in functions by type such as Aggregate, Conversion, and so on.
- Groups custom functions under the Custom node.
- Groups imported functions and stored procedures under the name of the datastore used to import them.

2. When you find the applicable function, select it in the library pane.

The software displays a description and the syntax of the function in the lower portion of the library pane.

3. Add the function to the editor pane by double-clicking it in the library pane.

19.9.4 Searching for a function

Use the search tools in the Smart Editor to search for a specific function.

1. In the editor pane, select the position in the string to place the function.
2. Open the *Functions* tab in the library pane and select the *Find* node.

The software displays a text box with a dropdown arrow. The dropdown arrow opens a list of terms for which you previously searched.

3. Enter a string in the text box. For example, enter **100**.
4. Press **Enter** or **Tab**.

The software lists all functions that contain the string under the Find node.

i Note

As you select each function, notice the description changes in the lower portion of the library pane.

5. Double-click the applicable function to place it into the editor.

You can also place the function in the following ways:

- drag-and-drop
- right-click on the function and press **Enter**
- select the function and press **Enter** or **Tab**

19.9.5 Validating in Smart Editor

The Smart Editor contains a validation tool that helps you correct syntax when there are errors.

If your expression is ready to be validated, and it is complete, the Smart Editor makes the [Validate](#) icon available for you. If the software does not enable the [Validate](#) option, close the Smart Editor and validate the expression in the Designer data flow using the Debug menu.

1. Select the [Validate](#) icon in the tool bar or right-click and select [Validate](#) from the dropdown list.



The software lists any errors in a separate pane below the editor.

2. Double-click each error from the list.
The editor redraws to show you where the error occurred in your text.
3. Fix the error and continue clicking the errors until you have fixed all of the errors.
4. Revalidate the expression to ensure that you have fixed all errors correctly.

Related Information

[Validate \[page 811\]](#)

20 Nested Data

The software maps nested data to a separate schema implicitly related to a single row and column of the parent schema. This mechanism is called Nested Relational Data Modelling (NRDM).

Real-world data often has hierarchical relationships that are represented in a relational database with master-detail schemas using foreign keys to create the mapping. However, some data sets, such as XML documents and SAP ERP IDocs, handle hierarchical relationships through nested data.

NRDM provides a way to view and manipulate hierarchical relationships within data flow sources, targets, and transforms.

Sales orders are often presented using nesting: the line items in a sales order are related to a single header and are represented using a nested schema. Each row of the sales order data set contains a nested line item schema.

OrderNo	CustID	ShipTo1	ShipTo2	LineItems
9999	1001	123 State St	Town, CA	

Item	Qty	ItemPrice
001	2	10
002	4	5

20.1 Representing hierarchical data

You can represent the same hierarchical data in several ways.

Examples include:

Multiple rows in a single data set

Order data set

OrderNo	CustID	ShipTo1	ShipTo2	Item	Qty	ItemPrice
9999	1001	123 State St	Town, CA	001	2	10
9999	1001	123 State St	Town, CA	002	4	5

Multiple data sets related by a join

Order header data set

OrderNo	CustID	ShipTo1	ShipTo2
9999	1001	123 State St	Town, CA

Line-item data set

OrderNo	Item	Qty	ItemPrice
9999	001	2	10
9999	002	4	5

WHERE Header.OrderNo=LineItem.OrderNo

Nested data

Using the nested data method can be more concise (no repeated information), and can scale to present a deeper level of hierarchical complexity. For example, columns inside a nested schema can also contain columns. There is a unique instance of each nested schema for each row at each level of the relationship.

Order data set

OrderNo	CustID	ShipTo1	ShipTo2	LinelItems
9999	1001	123 State St	Town, CA	

Item	Qty	ItemPrice
001	2	10
002	4	5

Generalizing further with nested data, each row at each level can have any number of columns containing nested schemas.

Order data set

OrderNo	LineItem	CustInfo
OrderNo		

Item	ItemQty	ItemPrice
001	2	
002	4	

Type	CustID	Contacts
Ship	1001	
Bill	7777	

Qty	Sell Price
1	200
10	190

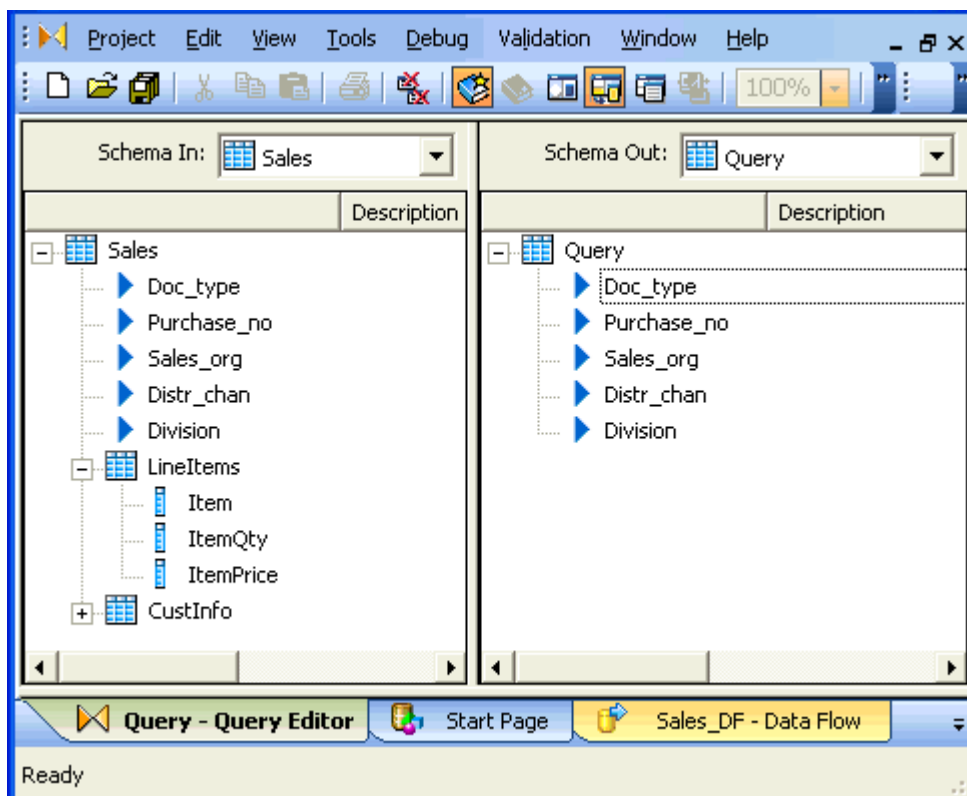
Qty	Sell Price
1	25
20	23

Name	Phone
Alvarez	555-1234
Tanaka	555-6666

Name	Phone
Samp	333-1234

You can see the structure of nested data in the input and output schemas of sources, targets, and transforms in data flows. Nested schemas appear with a schema icon paired with a plus sign, which indicates that the object contains columns. The structure of the schema shows how the data is ordered.

- Sales is the top-level schema.
- LineItems is a nested schema. The minus sign in front of the schema icon indicates that the column list is open.
- CustInfo is a nested schema with the column list closed.



20.2 Formatting XML documents

The software allows you to import and export metadata for XML documents (files or messages), which you can use as sources or targets in jobs. XML documents are hierarchical.

Their valid structure is stored in separate format documents.

The format of an XML file or message (.xml) can be specified using either an XML Schema (for example, .xsd) or a document type definition (.dtd).

When you import a format document's metadata, it is structured into the software's internal schema for hierarchical documents which uses the nested relational data model (NRDM).

Related Information

[XML Schema specification \[page 654\]](#)

[Mapping optional schemas \[page 667\]](#)

[Specifying source options for XML files \[page 666\]](#)

[Using Document Type Definitions \(DTDs\) \[page 668\]](#)

[Generating DTDs and XML Schemas from an NRDM schema \[page 673\]](#)

20.2.1 XML Schema specification

The software supports WC3 XML Schema Specification 1.0.

For an XML document that contains information to place a sales order—order header, customer, and line items—the corresponding XML Schema includes the order structure and the relationship between data.

Message with data

OrderNo	CustID	ShipTo1	ShipTo2	LineItems		
9999	1001	123 State St	Town, CA	Item	ItemQty	ItemPrice
				001	2	10
				002	4	5

Each column in the XML document corresponds to an ELEMENT or attribute definition in the XML schema.

Corresponding XML schema

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Order">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="OrderNo" type="xs:string" />
        <xs:element name="CustID" type="xs:string" />
        <xs:element name="ShipTo1" type="xs:string" />
        <xs:element name="ShipTo2" type="xs:string" />
        <xs:element maxOccurs="unbounded" name="LineItems">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Item" type="xs:string" />
              <xs:element name="ItemQty" type="xs:string" />
              <xs:element name="ItemPrice" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

Related Information

[Reference Guide: Objects, Descriptions of objects, XML schema](#)

20.2.2 About importing XML schemas

Import the metadata for each XML Schema you use.

The object library lists imported XML Schemas in the Nested Schemas category of the *Formats* tab.

When importing an XML Schema, the software reads the defined elements and attributes, and then imports the following:

- Document structure
- Namespace
- Table and column names
- Data type of each column
- Content type of each column
- Nested table and column attributes

While XML Schemas make a distinction between elements and attributes, the software imports and converts them all to nested table and column attributes.

Related Information

[Reference Guide: Objects, Descriptions of objects, XML schema](#)

20.2.2.1 Supported XML Schema components

SAP Data Services supports nearly all valid XML schemas.

The software imports and accepts XML schema features such as abstract types and blocking without issuing errors. The software also imports XML schema data types as well as elements and attribute names and their structure.

Once imported, double-click an XML schema format from the object library to view table and column names and structure. From the XML Format editor, right-click a column name and select edit properties, attributes, and data types.

Related Information

[Unsupported XML schema components \[page 656\]](#)

20.2.2.2 Unsupported XML schema components

There are several XML schema components that SAP Data Services does not support.

When you import XML schemas, the software ignores and does not import elements and attributes that it does not support.

Unsupported XML schema components

Component	Description
Annotation	The software ignores and does not import documentation and appinfo annotation components.
Non-native attributes	The software ignores and does not import non native attributes. Non native attributes come from a name space other than the one the XML schema uses. Even though the W3C XML Schema standard enables users to add non native attributes to attributes and elements, the software does not support them.
XDR files	The software ignores and does not import XDR (XML Data Reduced) files. XDR files were used before the W3C standards. Use a third-party tool to convert XDR to XML schema to enable importing an XDR file component

20.2.2.3 Included XML Schemas

An XML Schema can be extended by including pointers to other XML Schema files. This is done by using `<import>`, `<include>` and `<redefine>`. These elements are defined at the schema level.

The difference between `<include>` and `<import>` is that for `<include>` the name spaces must be identical in both XML Schemas. `<Redefine>` is similar to `<include>` except the caller can redefine one or more components in the related XML Schema.

When you import an XML Schema, SAP Data Services follows the links to included files to define additional metadata. The included schema information is saved in the repository so that at run time there is no need to access these files again. Inclusions can be files or URLs.

20.2.2.4 Groups

XML Schemas allow you to group elements and then refer to the group. A similar concept is available for attributes (called an attribute group). In SAP Data Services any reference to a group will be replaced by the contents of that group.

20.2.2.5 Rules for importing XML Schemas

SAP Data Services applies the following rules to convert an XML Schema to the software's internal schema:

1. Any element that contains an element only and no attributes becomes a column.
2. Any element with attributes or other elements becomes a table.
3. An attribute becomes a column in the table corresponding to the element it supports.
4. Any occurrence of `<choice>`, `<sequence>` or `<all>` uses the ordering given in the XML Schema as the column ordering in the internal data set.
5. Any occurrence of `<maxOccurs,>` from greater than 1 to "unbounded", becomes a table with an internally generated name (an implicit table).
The internally generated name is the name of the parent followed by an underscore, then the string "nt" followed by a sequence number. The sequence number starts at 1 and increments by 1.

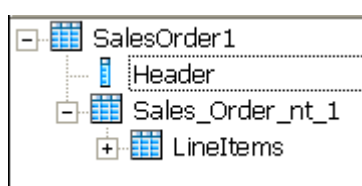
After applying these rules, the software uses two additional rules, except where doing so would allow more than one row for a root element:

1. If an implicit table contains one and only one nested table, then the implicit table can be eliminated and the nested table can be attached directly to the parent of the implicit table.

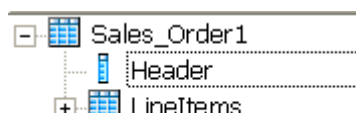
For example, the SalesOrder element might be defined as follows in an XML Schema:

```
<xs:element name="SalesOrder">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Header"/>
      <xs:element name="LineItems" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

When converted in the software, the LineItems element with MaxOccurs = "unbounded" would become an implicit table under the SalesOrder table. The LineItems element itself would be a nested table under the implicit table.



Because the implicit table contains one and only one nested table, the format would remove the implicit table.

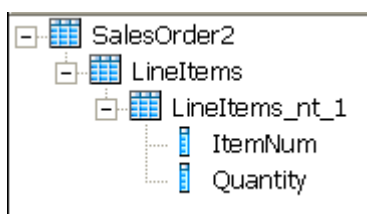


2. If a nested table contains one and only one implicit table, then the implicit table can be eliminated and its columns placed directly under the nested table.

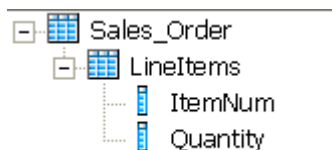
For example, the nested table LineItems might be defined as follows in an XML Schema:

```
<xs:element name="SalesOrder">
  <xs:element name="LineItems" minOccurs="0"
maxOccurs="unbounded">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="ItemNum"/>
        <xs:element ref="Quantity"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

When converted into the software, the grouping with MaxOccurs ="unbounded" would become an implicit table under the LineItems table. The ItemNum and Quantity elements would become columns under the implicit table.



Because the LineItems nested table contained one and only one implicit table, the format would remove the implicit table.



20.2.2.6 Limitations

If an XML schema definition contains the following elements or attributes, SAP Data Services imports it with the following limitations:

- Any element or anyAttribute
You can import an XML schema that contains an Any element or anyAttribute or both, but the format that the software creates does not show the Any element or anyAttribute. Consequently, the software ignores the content of the Any element or anyAttribute when it reads an XML instance document. When an element has type anyType, the software treats everything within it as a string and does not recognize the subelements within it.
- Mixed content
The structure of an XML schema usually consists of elements that contain subelements, and the subelements at the lowest level contain character data. However, an XML schema definition allows character data to appear next to subelements, and the character data is not confined to the lowest level. For instance documents that contain mixed content, the software ignores the character data between any two subcolumns, but captures the values of the subcolumns.

20.2.2.7 Importing an XML Schema

1. From the object library, click the *Format* tab.
2. Right-click the *Nested Schemas* icon and select ► *New* ► *XML Schema* ►.
3. Enter the settings for the XML schemas that you import based on the option descriptions below.

Option	Description
<i>Format name</i>	Enter the name that you want to use for the format in the software.
<i>File name / URL</i>	Enter or browse for the file name of the XML Schema or its URL address. i Note If your Job Server is on a different computer than the Data Services Designer, you cannot use <i>Browse</i> to specify the file path. You must type the path. You can type an absolute path or a relative path, but the Job Server must be able to access it.
<i>Namespace</i>	Select an imported XML schema from the drop-down list only if the root element name is not unique within the XML Schema. i Note When you import an XML Schema for a real-time web service job, use a unique target namespace for the schema. When Data Services generates the WSDL file for a real-time job with a source or target schema that has no target namespace, it adds an automatically generated target namespace to the types section of the XML schema. This can reduce performance because Data Services must suppress the namespace information from the web service request during processing, and then reattach the proper namespace information before returning the response to the client.
<i>Root element name</i>	Select the name of the primary node that you want to import from the drop-down list. The software only imports elements of the XML Schema that belong to this node or any subnodes.
<i>Circular level</i>	Specify the number of levels only if the XML Schema contains recursive elements (element A contains B, element B contains A). This value must match the number of recursive levels in the XML Schema's content. Otherwise, the job that uses this XML Schema will fail.
<i>Default varchar size</i>	Set the software to import strings as a varchar of any size. The default is 1024.

4. Click *OK*.

After you import an XML Schema, you can edit its column properties such as data type using the [General](#) tab of the Column Properties window. You can also view and edit nested table and column attributes from the Column Properties window.

20.2.2.8 Data type mappings

SAP Data Services imports data types for XML Schema elements and attributes.

There are two types of built-in data types, the Primitive data types and the Derived data types (derived from primitive). Each data type has the following values defined: space, lexical space, and constraining facet.

If the constraining facet `<length>` is missing when metadata is imported into the software, the default `varchar(1024)` is applied. Similarly, for a decimal the default values 28 and 2 are applied for `<precision>` and `<scale>`. All other facets like `<minInclusive>`, `<maxInclusive>`, `<minLength>` are imported as column attributes. Enumeration values are also imported as column attributes.

20.2.2.9 Primitive types

The following table lists Primitive XML Schema types, examples, and the corresponding data type in SAP Data Services. The constraining facets used are shown in bold.

XML Schema type	Example	Data type
AnyURI	<code>http://www.example.com/</code>	<code>Varchar(len) : len = <i>length</i> (in chars)</code>
Base64Binary	<code>GpM7</code>	<code>Varchar(len) : len = <i>length</i> (in octets)</code>
Boolean	<code>{true, false, 0, 1}</code>	<code>Varchar(5)</code>
Date	<code>CCYY-MM-DD</code>	<code>Datetime</code>
DateTime	<code>Format = CCYY-MM-DD HH:MM:SS</code>	<code>Datetime</code>
Decimal	<code>7.999</code>	<code>Decimal(p, s) : p = <i>totalDigits</i>, a maximum of 28 and s = <i>fractionDigits</i>, default = 28,2</code>
Double	<code>64 bit floating point</code>	<code>Double (In the software there is no difference between real and double)</code>
Duration	<code>P1Y2M3DT10H30M</code>	<code>Varchar(64)</code>
Float	<code>32 bit floating point, 12.78e-2</code>	<code>Real</code>
gDay		<code>Varchar(12)</code>
gMonth		<code>Varchar(12)</code>
gMonthDay		<code>Varchar(12)</code>
gYear		<code>Varchar(12)</code>
gYearMonth	<code>Gregorian CCYY-MM</code>	<code>Varchar(12)</code>
HexBinary	<code>0FB7</code>	<code>Varchar(len) : len = <i>length</i> (in octets)</code>

XML Schema type	Example	Data type
Notation		N/A
Qname	po:USAddress	Varchar(len) : len = <i>length</i> (in chars)
String	"Hello World"	Varchar(len) : len = <i>length</i> (in characters)
Time	HH:MM:SS	Datetime

20.2.2.10 Derived types

The following table lists pre-defined Derived XML Schema types, examples, and the corresponding data type in SAP Data Services. The constraining facets used are shown in bold.

XML Schema type	Example	Data type
NormalizedString	[No tab/CR/LF in string]	Varchar(len) : len = <i>length</i> (in characters)
Token		Varchar(len) : len = <i>length</i> (in characters)
Language	En-GB, en-US, fr	Varchar(len): len = <i>length</i> (in characters)
NMTOKEN	US, Brésil	Varchar(len): len = <i>length</i> (in characters)
NMTOKENS	Brésil Canada Mexique	Varchar(len): len = <i>length</i> (in characters)
Name	ShipTo	Varchar(len): len = <i>length</i> (in characters)
NCName	USAddress	Varchar(len): len = <i>length</i> (in characters)
ID		Varchar(len): len = <i>length</i> (in characters)
IDREF		Varchar(len): len = <i>length</i> (in characters)
IDREFS		Varchar(len): len = <i>length</i> (in characters)
ENTITY		Varchar(len): len = <i>length</i> (in characters)
ENTITIES		Varchar(len): len = <i>length</i> (in characters)
Integer		Int
NonPositiveInteger		Int
NegativeInteger		Int

XML Schema type	Example	Data type
Long		Decimal 28,0
Int		Int
Short		Int
Byte		Int
NonNegativeInteger		Int
UnsignedLong		Long
UnsignedShort		Int
UnsignedByte		Int
PositiveInteger		Int
AnyType (ur-type)	unspecified type	Varchar(255)

20.2.2.11 User-defined types

User-defined types are XML Schema attributes with a non-XML Schema name space. The XML Schema W3C standard uses a `SimpleType` element for a user-defined type.

When SAP Data Services finds a user-defined type it finds the base type and uses it to assign a data type for the element. For example: If element X has type `TelephoneNumber`, its type in the software is `varchar(8)`.

Some simple types are based on other simple types. In such cases the software traces back to the base type.

20.2.2.12 List types

XML Schemas have list types. When it encounters a list, SAP Data Services makes that list's corresponding data type a `varchar(1024)`. All the elements of the list are placed in the value of that column as a string (exactly as it is represented in the XML).

20.2.2.13 Union types

A union type enables an attribute or element value to be one or more instances of one type drawn from the union of multiple primitive type and list types. When it encounters a union, SAP Data Services makes that union's corresponding data type a `varchar(1024)`.

20.2.2.14 Viewing and editing nested table and column attributes for XML Schema

1. From the object library, select the *Formats* tab.
2. Expand the *Nested Schema* category.
3. Double-click an XML Schema name.

The *XML Format Editor* window opens in the workspace. The *XML Schema Format* portions are in the upper pane of the window.

The Type column displays the data types that the software uses when it imports the XML document metadata.

4. Double-click a nested table or column and select the *Attributes* tab of the *Properties* dialog to view or edit XML Schema attributes.

Related Information

[Reference Guide: Objects, Descriptions of objects, XML schema](#)

20.2.3 Support for abstract datatypes

SAP Data Services uses abstract data types to force substitution for a particular element or type.

The software supports XML file or message targets that use an XML schema that contains elements with abstract datatypes. To generate valid XML output with abstract datatypes, set the correct value for the `xsi:type` attribute. When you work abstract datatypes, ensure that you know which of the derived types are correct for any given element.

Note

By default, all elements with abstract datatypes have an attribute called `xsi:type`.

When you use XML schemas with namespaces, ensure that you include the right namespace in the type name. Obtain the right namespace tag by reviewing the namespace tags generated by Data Services, typically `ns1`, `ns2`, and so on. Then make sure that you use the tag that represents the right namespace in which the type exists.

Example

Assume that you have an element called **Publication**, which has an abstract type called **PublicationType**. When the software imports **Publication**, it adds an extra column named **xsi:type** as a child of **Publication**. Set the expression for this column to be equal to the expected type of the result. For example, it could be `BookType`. To add the correct tag, you execute your job and note the generated tag names. The generated tag name is **ns1** for a namespace named `<http://www.bookworld.com/>`. The expression of **xsi:type** is **ns1:BookType**.

20.2.3.1 Import abstract data types

When you import an abstract data type element, there are specific requirements that you should know about.

Ensure that you prepare for importing abstract data types in the following ways:

- When an element is defined as abstract, ensure that a member of the element substitution group appears in the instance document.
- When a type is defined as abstract, ensure that the instance document uses a type derived from it, identified by the `xsi:type` attribute.

❖ Example

An abstract element named `PublicationType` has a substitution group that consists of complex types such as `MagazineType`, `BookType`, and `NewspaperType`. The software default behavior is to select all complex types in the substitution group or all derived types for the abstract type. However, you can choose to select a subset.

20.2.3.2 Limiting the number of derived types to import for an abstract type

1. In the *Import XML Schema Format* window, when you enter the file name or URL address of an XML Schema that contains an abstract type, the *Abstract type* button is enabled.

For example, the following excerpt from an `xsd` defines the `PublicationType` element as abstract with derived types `BookType` and `MagazineType`:

```
<xsd:complexType name="PublicationType" abstract="true">
  <xsd:sequence>
    <xsd:element name="Title" type="xsd:string"/>
    <xsd:element name="Author" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="Date" type="xsd:gYear"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="BookType">
  <xsd:complexContent>
    <xsd:extension base="PublicationType">
      <xsd:sequence>
        <xsd:element name="ISBN" type="xsd:string"/>
        <xsd:element name="Publisher" type="xsd:string"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="MagazineType">
  <xsd:complexContent>
    <xsd:restriction base="PublicationType">
      <xsd:sequence>
        <xsd:element name="Title" type="xsd:string"/>
        <xsd:element name="Author" type="xsd:string"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="Date" type="xsd:gYear"/>
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
```

```
</xsd:complexType>
```

2. To select a subset of derived types for an abstract type, click the [Abstract type](#) button and take the following actions:
 - a. From the drop-down list on the [Abstract type](#) box, select the name of the abstract type.
 - b. Select the check boxes in front of each derived type name that you want to import.
 - c. Click [OK](#).

i Note

When you edit your XML schema format, the software selects all derived types for the abstract type by default. In other words, the subset that you previously selected is not preserved.

20.2.4 Importing substitution groups

An XML schema uses substitution groups to assign elements to a special group of elements that can be substituted for a particular named element called the head element.

The list of substitution groups can have hundreds or even thousands of members, but an application typically only uses a limited number of them. The default is to select all substitution groups, but you can choose to select a subset.

20.2.5 Limiting the number of substitution groups to import

You can select a subset of substitution groups to import.

1. In the [Import XML Schema Format](#) window, when you enter the file name or URL address of an XML Schema that contains substitution groups, the [Substitution Group](#) button is enabled.

For example, the following excerpt from an xsd defines the PublicationType element with substitution groups MagazineType, BookType, AdsType, and NewspaperType:

```
<xsd:element name="Publication" type="PublicationType"/>
  <xsd:element name="BookStore">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Publication" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Magazine" type="MagazineType"
    substitutionGroup="Publication"/>
  <xsd:element name="Book" type="BookType" substitutionGroup="Publication"/>
  <xsd:element name="Ads" type="AdsType" substitutionGroup="Publication"/>
  <xsd:element name="Newspaper" type="NewspaperType"
    substitutionGroup="Publication"/>
```

2. Click the [Substitution Group](#) button.
 - a. From the drop-down list on the [Substitution group](#) box, select the name of the substitution group.
 - b. Select the check boxes in front of each substitution group name that you want to import.

- c. Click [OK](#).

i Note

When you edit your XML schema format, the software selects all elements for the substitution group by default. In other words, the subset that you previously selected is not preserved.

20.2.6 Specifying source options for XML files

Create a data flow with an XML file as source.

After you import metadata for XML documents (files or messages), you create a data flow to use the XML documents as sources or targets in jobs.

Follow these steps to create a data flow with a source XML file:

1. From the [Nested Schema](#) category in the [Format](#) tab of the object library, drag the XML Schema that defines your source XML file into your data flow.
2. Place a query in the data flow and connect the XML source to the input of the query.
3. Double-click the XML source in the work space to open the XML Source File Editor.
4. Specify the name of the source XML file in the [XML file](#) text box.

Related Information

[Reading multiple XML files at one time \[page 666\]](#)

[Identifying source file names \[page 224\]](#)

[Reference Guide: Objects, Descriptions of objects, Source, XML file source](#)

20.2.6.1 Reading multiple XML files at one time

The software can read multiple files with the same format from a single directory using a single source object.

Set up a data flow as discussed in [Specifying source options for XML files \[page 666\]](#).

1. Double-click the source XML file in the applicable data flow to open the [Nested Schemas Source File Editor](#) window in the Workspace.
2. In the [Source](#) tab in the lower portion of the window, enter an applicable file name that includes a wild card character (* or ?) in the [File](#) option.

For example:

D:\orders\1999?????.xml reads all .xml files that contain “1999” in the file name from the specified directory.

D:\orders*.xml reads all files with the .xml extension from the specified directory.

Related Information

[Reference Guide: Objects, Descriptions of objects, Source, XML file source](#)

20.2.6.2 Identifying source file names

You might want to identify the source XML file for each row in your source output in the following situations:

- You specified a wildcard character to read multiple source files at one time.
- You load from a different source file on different days.

20.2.6.2.1 Identifying the source XML file for each row in the target

Set up a data flow as discussed in [Specifying source options for XML files \[page 666\]](#).

1. Double-click the source XML file in the applicable data flow to open the *Nested Schemas Source File Editor* window in the Workspace.
2. In the *Source* tab in the lower portion of the window, select *Include file name column*.

This option generates a column named DI_FILENAME to contain the name of the source XML file.

3. Open the Query transform in the data flow. In the Query editor, map the DI_FILENAME column from Schema In to Schema Out.

When you run the job, the target DI_FILENAME column will contain the source XML file name for each row in the target.

20.2.7 Mapping optional schemas

You can quickly specify default mapping for optional schemas without having to manually construct an empty nested table for each optional schema in the Query transform. Also, when you import XML schemas (either through DTDs or XSD files), the software automatically marks nested tables as optional if the corresponding option was set in the DTD or XSD file. The software retains this option when you copy and paste schemas into your Query transforms.

This feature is especially helpful when you have very large XML schemas with many nested levels in your jobs. When you make a schema column optional and do not provide mapping for it, the software instantiates the empty nested table when you run the job.

While a schema element is marked as optional, you can still provide a mapping for the schema by appropriately programming the corresponding sub-query block with application logic that specifies how the software should produce the output. However, if you modify any part of the sub-query block, the resulting query block must be complete and conform to normal validation rules required for a nested query block. You must map any output schema not marked as optional to a valid nested query block. The software generates a NULL in the

corresponding PROJECT list slot of the ATL for any optional schema without an associated, defined sub-query block.

20.2.7.1 Making a nested table "optional"

1. Right-click a nested table and select *Optional* to toggle it on. To toggle it off, right-click the nested table again and select *Optional* again.
2. You can also right-click a nested table and select *Properties*, and then open the Attributes tab and set the *Optional Table* attribute value to *yes* or *no*. Click *Apply* and *OK* to set.

i Note

If the Optional Table value is something other than yes or no, the nested table cannot be marked as optional.

When you run a job with a nested table set to optional and you have nothing defined for any columns and nested tables beneath that table, the software generates special ATL and does not perform user interface validation for this nested table.

Example:

```
CREATE NEW Query ( EMPNO int KEY ,
ENAME varchar(10),
JOB varchar (9)
NT1 al_nested_table ( DEPTNO int KEY ,
DNAME varchar (14),
NT2 al_nested_table (C1 int) ) SET("Optional
Table"= 'yes') )
AS SELECT EMP.EMPNO, EMP.ENAME, EMP.JOB,
NULL FROM EMP, DEPT;
```

i Note

You cannot mark top-level schemas, unnested tables, or nested tables containing function calls optional.

20.2.8 Using Document Type Definitions (DTDs)

The format of an XML document (file or message) can be specified by a document type definition (DTD). The DTD describes the data contained in the XML document and the relationships among the elements in the data.

For an XML document that contains information to place a sales order—order header, customer, and line items—the corresponding DTD includes the order structure and the relationship between data.

Message with data

OrderNo	CustID	ShipTo1	ShipTo2	LineItems		
9999	1001	123 State St	Town, CA	Item	ItemQty	ItemPrice
				001	2	10
				002	4	5

Each column in the XML document corresponds to an ELEMENT definition.

Corresponding DTD Definition

```
<?xml encoding="UTF-8"?>
<!ELEMENT Order (OrderNo, CustID, ShipTo1, ShipTo2, LineItems+)>
<!ELEMENT OrderNo (#PCDATA)>
<!ELEMENT CustID (#PCDATA)>
<!ELEMENT ShipTo1 (#PCDATA)>
<!ELEMENT ShipTo2 (#PCDATA)>
<!ELEMENT LineItems (Item, ItemQty, ItemPrice)>
<!ELEMENT Item (#PCDATA)>
<!ELEMENT ItemQty (#PCDATA)>
<!ELEMENT ItemPrice (#PCDATA)>
```

Import the metadata for each DTD you use. The object library lists imported DTDs in the [Formats](#) tab.

You can import metadata from either an existing XML file (with a reference to a DTD) or DTD file. If you import the metadata from an XML file, the software automatically retrieves the DTD for that XML file.

When importing a DTD, the software reads the defined elements and attributes. The software ignores other parts of the definition, such as text and comments. This allows you to modify imported XML data and edit the data type as needed.

Related Information

[Reference Guide: Objects, Descriptions of objects, DTD](#)

20.2.8.1 Rules for importing DTDs

SAP Data Services uses conversion rules to convert a DTD to an internal schema.

SAP Data Services applies the following rules to convert a DTD to an internal schema:

- Any element that contains an PCDATA only and no attributes becomes a column.
- Any element with attributes or other elements (or in mixed format) becomes a table.

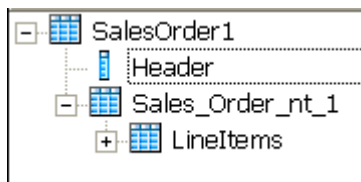
- An attribute becomes a column in the table corresponding to the element it supports.
- Any occurrence of choice (.) or sequence (|) operators uses the ordering given in the DTD as the column ordering in the internal data set.
- Any occurrence of a multiple entities, such as (*) or (+), becomes a table with an internally generated name (an implicit table).
- The internally generated name is the name of the parent followed by an underscore, then the string "nt" followed by a sequence number. The sequence number starts at 1 and increments by 1.

After applying these rules, the software uses two additional rules, except where doing so would allow more than one row for a root element:

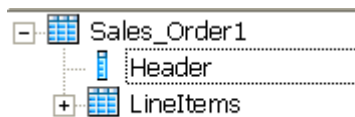
- If an implicit table contains one and only one nested table, then the implicit table can be eliminated and the nested table can be attached directly to the parent of the implicit table.
For example, the SalesOrder element might be defined as follows in a DTD:

```
<!ELEMENT SalesOrder (Header, LineItems*)>
```

When converted into the software, the LineItems element with the zero or more operator would become an implicit table under the SalesOrder table. The LineItems element itself would be a nested table under the implicit table.



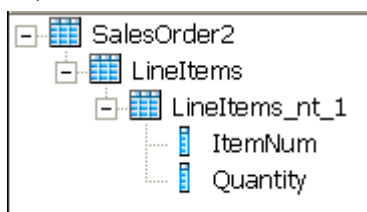
Because the implicit table contains one and only one nested table, the format would remove the implicit table.



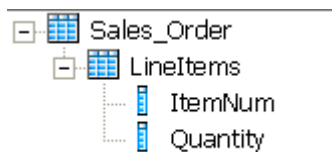
- If a nested table contains one and only one implicit table, then the implicit table can be eliminated and its columns placed directly under the nested table.
For example, the nested table LineItems might be defined as follows in a DTD:

```
<!ELEMENT LineItems (ItemNum, Quantity)*>
```

When converted into the software, the grouping with the zero or more operator would become an implicit table under the LineItems table. The ItemNum and Quantity elements would become columns under the implicit table.



Because the LineItems nested table contained one and only one implicit table, the format would remove the implicit table.



20.2.8.2 Importing a DTD or XML Schema format

1. From the object library, click the *Format* tab.
2. Right-click the *Nested Schemas* icon and select **New > DTD**.
3. Enter settings into the Import DTD Format window:
 - In the *DTD definition name* box, enter the name that you want to give the imported DTD format in the software.
 - Enter the file that specifies the DTD you want to import.

Note

If your Job Server is on a different computer than the Designer, you cannot use Browse to specify the file path. You must type the path. You can type an absolute path or a relative path, but the Job Server must be able to access it.

- If importing an XML file, select *XML* for the *File type* option. If importing a DTD file, select the *DTD* option.
 - In the *Root element name* box, select the name of the primary node that you want to import. The software only imports elements of the DTD that belong to this node or any subnodes.
 - If the DTD contains recursive elements (element A contains B, element B contains A), specify the number of levels it has by entering a value in the *Circular level* box. This value must match the number of recursive levels in the DTD's content. Otherwise, the job that uses this DTD will fail.
 - You can set the software to import strings as a varchar of any size. Varchar 1024 is the default.
4. Click *OK*.

After you import a DTD, you can edit its column properties, such as data type, using the *General* tab of the *Column Properties* window. You can also view and edit DTD nested table and column attributes from the *Column Properties* window.

20.2.8.3 Design considerations

Use design techniques to improve performance and tune nested relational data models for DTDs.

The following areas provide opportunities for you to improve performance and tune the nested-relational data model results for a given DTD:

- **Recursion**
If the DTD contains an element that uses an ancestor element in its definition, SAP Data Services expands the definition of the ancestor for a fixed number of levels. For example, given the following definition of element "A":

A: B, C
 B: E, F
 F: A, H

The software produces a table for the element "F" that includes an expansion of "A." In this second expansion of "A," "F" appears again, and so on until the fixed number of levels. In the final expansion of "A," the element "F" appears with only the element "H" in its definition.

- Repeated column names
 The software does not allow more than one column with the same name at the same level in a table. If the internal schema that the software produces from a DTD contains duplicate names, the software adds a suffix to each instance of the duplicated name to ensure unique column names. The suffix is an underscore followed by a sequence number, starting at 1 and incrementing by 1.
 A DTD can produce duplicate names when the DTD contains a repeated element at one level or an element with a scalar value with an attribute of the same name.
- Ambiguous DTDs
 You can create a DTD such that the software does not have enough information to make a unique decision when generating the internal data set. The software reacts to an ambiguous DTD by throwing an error for the XML message source at run time. An example of an ambiguous definition is as follows:

DTD	A: ((B, (C) *) (B, (D) *)) +
Schema in Data Services	A: (B, (C) *, B, (D) *) *
XML input	A: text <D>1</D> <D>2</D>

The software will use the B element data to populate the first B column, then use the D element data to populate the D element. If this data is then translated back into XML, it would be invalid relative to the DTD.

Metadata

If you delete a DTD from the object library, XML file and message sources or targets that are based on this format are invalid. The software marks the source or target objects with an icon that indicates the calls are no longer valid.



To restore the invalid objects, you must delete the source or target and replace it with a source or target based on an existing DTD.

20.2.8.4 Viewing and editing nested table and column attributes for DTDs

- From the object library, select the *Formats* tab.

2. Expand the *Nested Schemas* category.
3. Double-click a DTD name.

The *DTD Format* window appears in the workspace.

4. Double-click a nested table or column.

The *Column Properties* window opens.

5. Select the *Attributes* tab to view or edit DTD attributes.

20.2.8.5 Error checking

Control whether SAP Data Services checks an incoming XML file or message for validity.

If you choose to check each XML file or message, the software uses the DTD imported and stored in the repository rather than a DTD specified by a given XML file or message. If a file or message is invalid relative to the DTD, the real-time job produces an error and shuts down.

During development, you might validate all files and messages to test for error conditions. During production, you might choose to accept rare invalid files or messages and risk ambiguous or incorrect data.

All files or messages that the software produces for an XML file or message target are validated against the imported DTD.

You can enable or disable validation for an XML file or message source or target in that object's editor.

20.2.9 Generating DTDs and XML Schemas from an NRDM schema

You can right-click any schema from within a query editor in the Designer and generate a DTD or an XML Schema that corresponds to the structure of the selected schema (either NRDM or relational).

This feature is useful if you want to stage data to an XML file and subsequently read it into another data flow.

1. Generate a DTD/XML Schema.
2. Use the DTD/XML Schema to setup an XML format.
3. Use the XML format to set up an XML source for the staged file.

The DTD/XML Schema generated will be based on the following information:

- Columns become either elements or attributes based on whether the XML Type attribute is set to ATTRIBUTE or ELEMENT.
- If the Required attribute is set to NO, the corresponding element or attribute is marked optional.
- Nested tables become intermediate elements.
- The Native Type attribute is used to set the type of the element or attribute.
- While generating XML Schemas, the MinOccurs and MaxOccurs values are set based on the Minimum Occurrence and Maximum Occurrence attributes of the corresponding nested table.

No other information is considered while generating the DTD or XML Schema.

Related Information

[Reference Guide: Objects, Descriptions of objects, DTD](#)
[Reference Guide: Objects, Descriptions of objects, XML schema](#)

20.3 Operations on nested data

This section discusses the operations that you can perform on nested data.

20.3.1 Overview of nested data and the Query transform

When working with nested data, the Query transform provides an interface to perform SELECT statements at each level of the relationship that you define in the output schema.

With relational data, a Query transform allows you to execute a SELECT statement. The mapping between input and output schemas defines the project list for the statement.

You use the Query transform to manipulate nested data. If you want to extract only part of the nested data, you can use the XML_Pipeline transform.

Without nested schemas, the Query transform assumes that the FROM clause in the SELECT statement contains the data sets that are connected as inputs to the query object. When working with nested data, you must explicitly define the FROM clause in a query. The software assists by setting the top-level inputs as the default FROM clause values for the top-level output schema.

The other SELECT statement elements defined by the query work the same with nested data as they do with flat data. However, because a SELECT statement can only include references to relational data sets, a query that includes nested data includes a SELECT statement to define operations for each parent and child schema in the output.

The Query Editor contains a tab for each clause of the query:

- SELECT provides an option to specify distinct rows to output (discarding any identical duplicate rows).
- FROM lists all input schemas and allows you to specify join pairs and conditions.

The parameters you enter for the following tabs apply only to the current schema (displayed in the Schema Out text box at the top right of the Query Editor):

- WHERE
- GROUP BY
- ORDER BY

Related Information

[Reference Guide: Transforms, Data Integrator transforms, XML_Pipeline](#)

20.3.2 FROM clause construction

The FROM clause allows you to specify the tables and views to use in a join statement.

The FROM clause is located at the bottom of the FROM tab. It automatically populates with the information included in the Input Schema(s) section at the top, and the Join Pairs section in the middle of the tab. You can change the FROM clause by changing the selected schema in the Input Schema(s) area and the Join Pairs section.

Schemas selected in the Input Schema(s) section (and reflected in the FROM clause), including columns containing nested schemas, are available to be included in the output.

When you include more than one schema in the Input Schema(s) section (by selecting the [From](#) check box), you can specify join pairs and join conditions as well as enter join rank and cache for each input schema.

FROM clause descriptions and the behavior of the query are exactly the same with nested data as with relational data. The current schema allows you to distinguish multiple SELECT statements from each other within a single query. However, because the SELECT statements are dependent upon each other, and because the user interface makes it easy to construct arbitrary data sets, determining the appropriate FROM clauses for multiple levels of nesting can be complex.

A FROM clause can contain:

- Any top-level schema from the input
- Any schema that is a column of a schema in the FROM clause of the parent schema
- Any join conditions from the join pairs

The FROM clause forms a path that can start at any level of the output. The first schema in the path must always be a top-level schema from the input.

The data that a SELECT statement from a lower schema produces differs depending on whether or not a schema is included in the FROM clause at the top-level.

The next two examples use the sales order data set to illustrate scenarios where FROM clause values change the data resulting from the query.

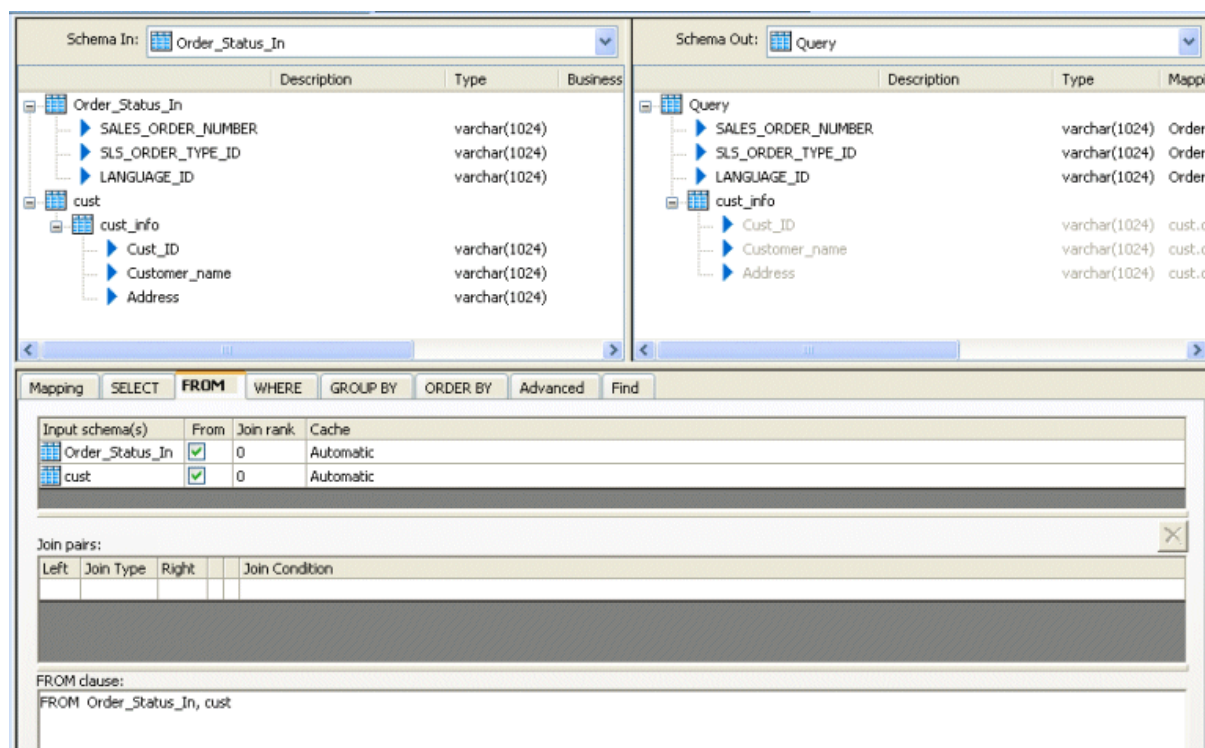
Related Information

[Reference Guide: Transforms, Query, Schema In and Schema Out \[page 620\]](#)

20.3.2.1 Example: FROM clause includes all top-level inputs

To include detailed customer information for all of the orders in the output, join the Order_Status_In schema at the top level with the Cust schema. Include both input schemas at the top level in the FROM clause to produce

the appropriate data. When you select both input schemas in the Input schema(s) area of the FROM tab, they automatically appear in the FROM clause.

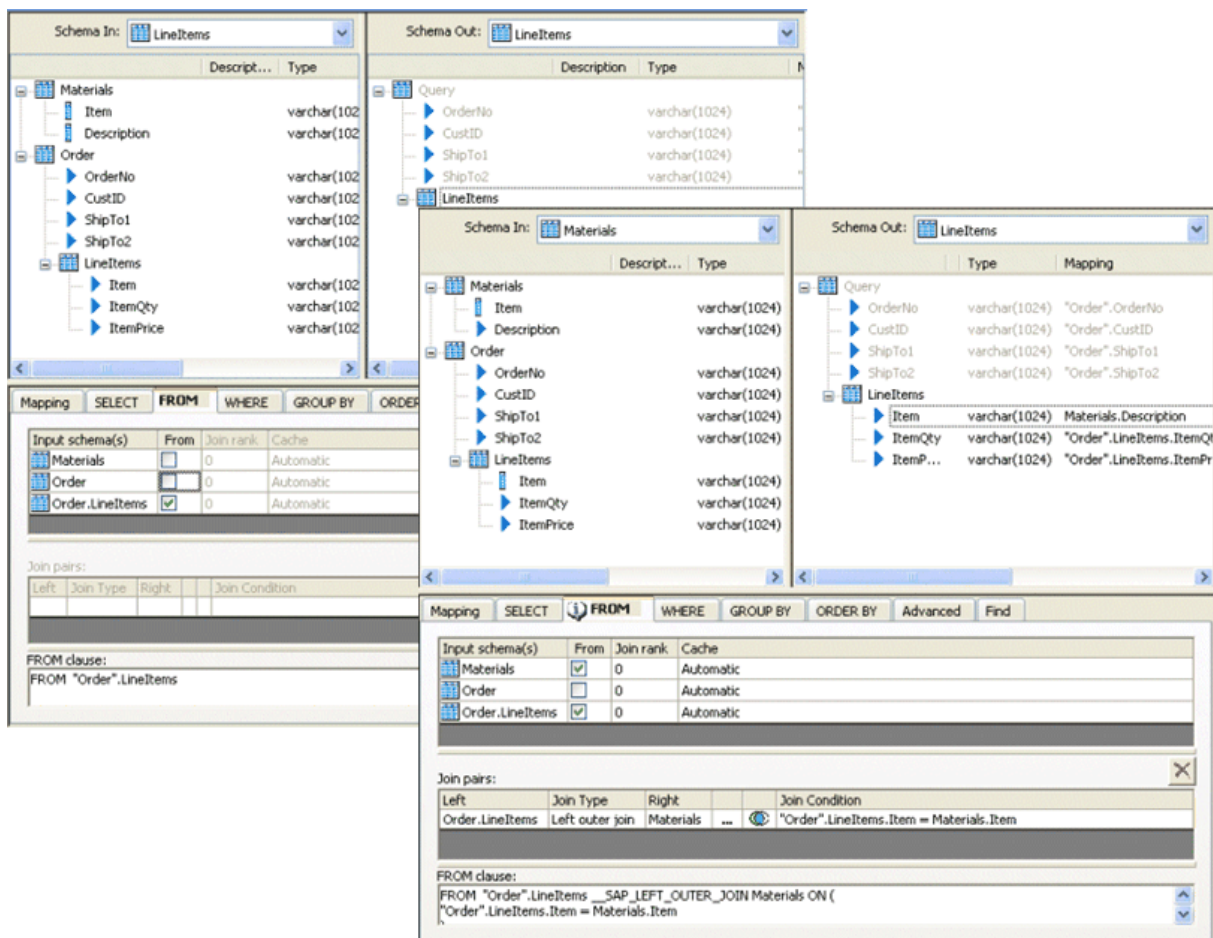


Observe the following points:

- The Input schema(s) table in the FROM tab includes the two top-level schemas Order_Status_In and Cust (this is also reflected in the FROM clause).
- The Schema Out pane shows the nested schema, cust_info, and the columns Cust_ID, Customer_name, and Address.

20.3.2.2 Example: Lower level FROM clause contains top-level input

Suppose you want the detailed information from one schema to appear for each row in a lower level of another schema. For example, the input includes a top-level Materials schema and a nested LinelItems schema, and you want the output to include detailed material information for each line item. The graphic below illustrates how to set this up in the Designer.



The example on the left shows the following setup:

- The Input Schema area in the FROM tab shows the nested schema LineItems selected.
- The FROM tab shows the FROM clause "FROM 'Order'.LineItems".

The example on the right shows the following setup:

- The Materials.Description schema is mapped to LineItems.Item output schema.
- The Input schema(s) Materials and Order.LineItems are selected in the Input Schema area in the FROM tab (the From column has a check mark).
- A Join Pair is created joining the nested Order.LineItems schema with the top-level Materials schema using a left outer join type.
- A Join Condition is added where the Item field under the nested schema LineItems is equal to the Item field in the top-level Materials schema.

The resulting FROM clause:

```
"Order".LineItems.Item = Materials.Item
```

20.3.3 Nesting columns

When you nest rows of one schema inside another, the data set produced in the nested schema is the result of a query against the first one using the related values from the second one.

For example, if you have sales-order information in a header schema and a line-item schema, you can nest the line items under the header schema. The line items for a single row of the header schema are equal to the results of a query including the order number:

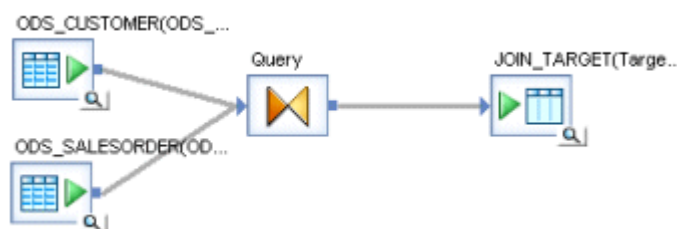
```
SELECT * FROM LineItems
WHERE Header.OrderNo = LineItems.OrderNo
```

You can use a query transform to construct a nested data set from relational data. When you indicate the columns included in the nested schema, specify the query used to define the nested data set for each row of the parent schema.

20.3.3.1 Constructing a nested data set

Follow the steps below to set up a nested data set.

1. Create a data flow with the input sources that you want to include in the nested data set.
2. Place a Query transform and a target table in the data flow. Connect the sources to the input of the query.



3. Open the Query transform and set up the *Select list*, *FROM clause*, and *WHERE clause* to describe the SELECT statement that the query executes to determine the top-level data set.

Option	Notes
<i>Select list</i>	Map the input schema items to the output schema by dragging the columns from the input schema to the output schema. You can also include new columns or include mapping expressions for the columns.
<i>FROM clause</i>	Include the input sources in the list on the <i>FROM</i> tab, and include any joins and join conditions required to define the data.
<i>WHERE clause</i>	Include any filtering required to define the data set for the top-level output.

4. Create a new schema in the output.

Right-click in the Schema Out area of the Query Editor, choose [New Output Schema](#), and name the new schema. A new schema icon appears in the output, nested under the top-level schema.

You can also drag an entire schema from the input to the output.

5. Change the current output schema to the nested schema by right-clicking the nested schema and selecting [Make Current](#).

The Query Editor changes to display the new current schema.

6. Indicate the FROM clause, [Select list](#), and [WHERE clause](#) to describe the [SELECT statement](#) that the query executes to determine the top-level data set.

Option	Notes
FROM clause	If you created a new output schema, you need to drag schemas from the input to populate the FROM clause. If you dragged an existing schema from the input to the top-level output, that schema is automatically mapped and listed in the From tab.
Select list	Only columns are available that meet the requirements for the FROM clause.
WHERE clause	Only columns are available that meet the requirements for the FROM clause.

7. If the output requires it, nest another schema at this level.

Repeat steps 4 through 6 in this current schema for as many nested schemas that you want to set up.

8. If the output requires it, nest another schema under the top level.

Make the top-level schema the current schema.

Related Information

[FROM clause construction \[page 675\]](#)

[Reference Guide: Transforms, Query](#)

[Reference Guide: Transforms, Query, Schema In and Schema Out \[page 620\]](#)

20.3.4 Using correlated columns in nested data

Correlation allows you to use columns from a higher-level schema to construct a nested schema.

In a nested-relational model, the columns in a nested schema are implicitly related to the columns in the parent row. To take advantage of this relationship, you can use columns from the parent schema in the construction of the nested schema. The higher-level column is a correlated column.

Including a correlated column in a nested schema can serve two purposes:

- The correlated column is a key in the parent schema. Including the key in the nested schema allows you to maintain a relationship between the two schemas after converting them from the nested data model to a relational model.

- The correlated column is an attribute in the parent schema. Including the attribute in the nested schema allows you to use the attribute to simplify correlated queries against the nested data.

To include a correlated column in a nested schema, you do not need to include the schema that includes the column in the FROM clause of the nested schema.

1. Create a data flow with a source that includes a parent schema with a nested schema.

For example, the source could be an order header schema that has a LinelItems column that contains a nested schema.

2. Connect a query to the output of the source.
3. In the query editor, copy all columns of the parent schema to the output.

In addition to the top-level columns, the software creates a column called LinelItems that contains a nested schema that corresponds to the LinelItems nested schema in the input.

4. Change the current schema to the LinelItems schema.
5. Include a correlated column in the nested schema.

Correlated columns can include columns from the parent schema and any other schemas in the FROM clause of the parent schema.

For example, drag the OrderNo column from the Header schema into the LinelItems schema. Including the correlated column creates a new output column in the LinelItems schema called OrderNo and maps it to the Order.OrderNo column. The data set created for LinelItems includes all of the LinelItems columns and the OrderNo.

If the correlated column comes from a schema other than the immediate parent, the data in the nested schema includes only the rows that match both the related values in the current row of the parent schema and the value of the correlated column.

20.3.5 Distinct rows and nested data

The *Distinct rows* option in Query transforms removes any duplicate rows at the top level of a join.

This is particularly useful to avoid cross products in joins that produce nested output.

20.3.6 Grouping values across nested schemas

When you specify a Group By clause for a schema with a nested schema, the grouping operation combines the nested schemas for each group.

For example, to assemble all the line items included in all the orders for each state from a set of orders, you can set the Group By clause in the top level of the data set to the state column (Order.State) and create an output schema that includes State column (set to Order.State) and LinelItems nested schema.

Order data set

OrderNo	CustID	State	LinItems
9999	1000	CA	
9999	1001	CA	
9777	1202	TX	

Item	ItemQty	ItemPrice
001	2	10
002	4	5

Item	ItemQty	ItemPrice
001	7	23
002	7	10

Item	ItemQty	ItemPrice
001	9	99
002	1	2

Order data set with GroupBy State

State	LinItems
CA	
TX	

Item	ItemQty	ItemPrice
001	2	10
002	4	5
001	7	23
002	7	10

Item	ItemQty	ItemPrice
001	9	99
002	1	2

The result is a set of rows (one for each state) that has the State column and the LinItems nested schema that contains all the LinItems for all the orders for that state.

20.3.7 Unnesting nested data

Loading a data set that contains nested schemas into a relational (non-nested) target requires that the nested rows be unnested.

For example, a sales order may use a nested schema to define the relationship between the order header and the order line items. To load the data into relational schemas, the multi-level must be unnested. Unnesting a schema produces a cross-product of the top-level schema (parent) and the nested schema (child).

Nested data set

OrderNo	CustID	LinItems
9999	1001	

Item	ItemQty
001	2
002	4

Header Schema

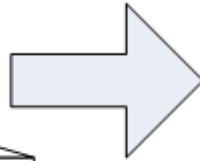
OrderNo	CustID	Item	ItemQty
9999	1001	001	2
9999	1001	002	4

It is also possible that you would load different columns from different nesting levels into different schemas. A sales order, for example, may be flattened so that the order number is maintained separately with each line item and the header and line item information loaded into separate schemas.

Nested data set

OrderNo	CustID	LineItems
9999	1001	

Item	ItemQty
001	2
002	4



Header schema

OrderNo	CustID
9999	1001

Line-item schema

OrderNo	Item	ItemQty
9999	001	2
9999	002	4

The software allows you to unnest any number of nested schemas at any depth. No matter how many levels are involved, the result of unnesting schemas is a cross product of the parent and child schemas. When more than one level of unnesting occurs, the inner-most child is unnested first, then the result—the cross product of the parent and the inner-most child—is then unnested from its parent, and so on to the top-level schema.

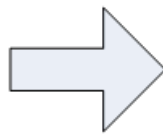
Order data set

OrderNo	LineItem
OrderNo	

Item	ItemQty	ItemPrice
001	2	
002	4	

Qty	Sell Price
1	25
20	23

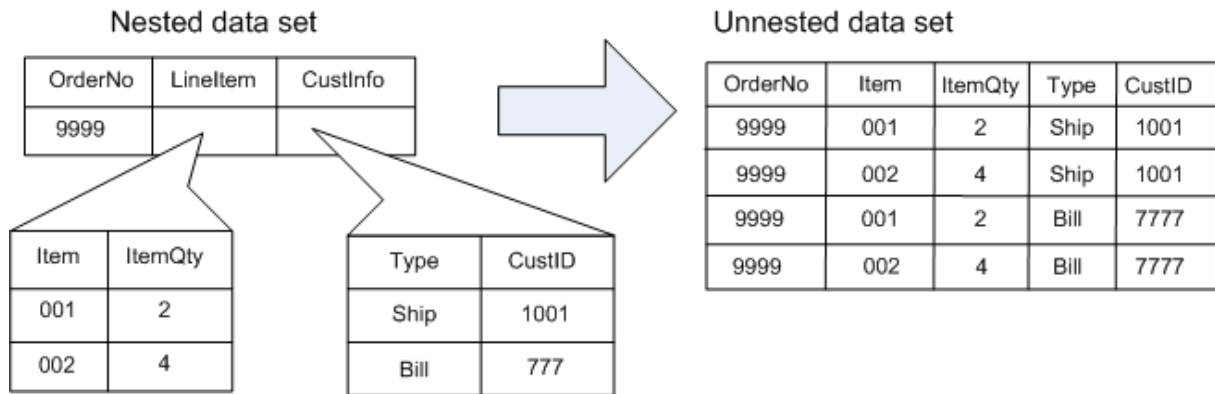
Qty	Sell Price
1	200
10	190



Unnested data set

OrderNo	CustID	Item	ItemQty	Qty	SellPrice
9999	1001	001	2	1	200
9999	1001	001	2	10	190
9999	1001	002	4	1	25
9999	1001	002	4	20	23

Unnesting all schemas (cross product of all data) might not produce the results that you intend. For example, if an order includes multiple customer values such as ship-to and bill-to addresses, flattening a sales order by unnesting customer and line-item schemas produces rows of data that might not be useful for processing the order.



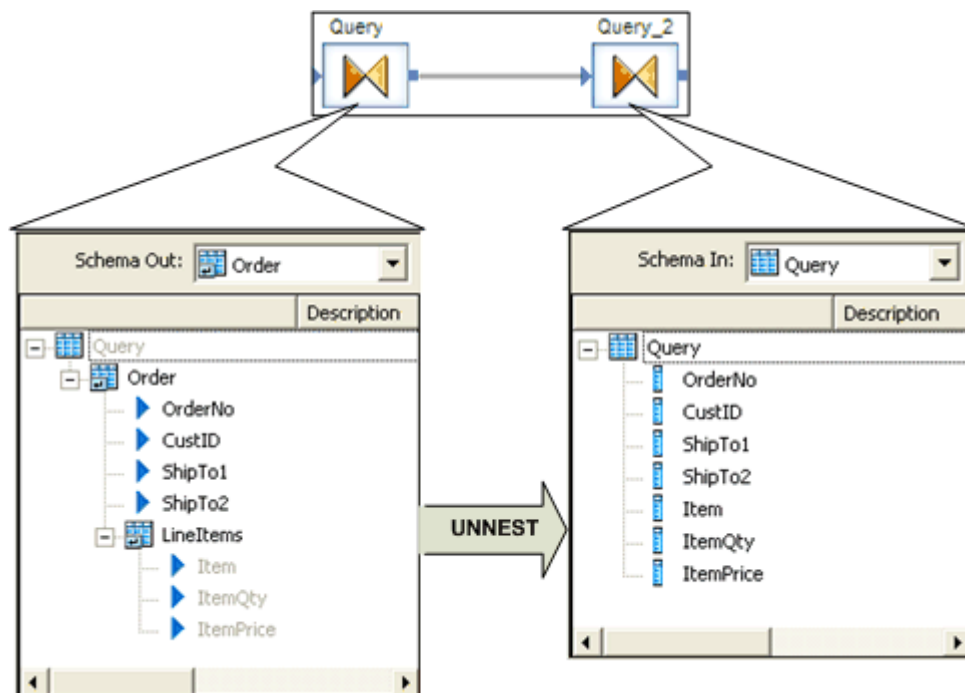
20.3.7.1 Unnesting nested data

1. Create the output that you want to unnest in the output schema of a query.

Data for unneeded columns or schemas might be more difficult to filter out after the unnesting operation. You can use the Cut command to remove columns or schemas from the top level; to remove nested schemas or columns inside nested schemas, make the nested schema the current schema, and then cut the unneeded columns or nested columns.

2. For each of the nested schemas that you want to unnest, right-click the schema name and choose *Unnest*.

The output of the query (the input to the next step in the data flow) includes the data in the new relationship, as the following diagram shows.



20.3.8 Transforming lower levels of nested data

Nested data included in the input to transforms (with the exception of a Query or XML_Pipeline transform) passes through the transform without being included in the transform's operation. Only the columns at the first level of the input data set are available for subsequent transforms.

20.3.8.1 Transforming values in lower levels of nested schemas

1. Take one of the following actions to obtain the nested data:
 - Use a Query transform to unnest the data.
 - Use an XML_Pipeline transform to select portions of the nested data.
 - Perform the transformation.
2. Nest the data again to reconstruct the nested relationships.

Related Information

[Unnesting nested data \[page 681\]](#)

[Reference Guide: Transforms, Data Integrator transforms, XML_Pipeline](#)

20.4 XML extraction and parsing for columns

You can use the software to extract XML data stored in a source table or flat file column, transform it as NRDM data, and then load it to a target or flat file column.

In addition, you can extract XML message and file data, representing it as NRDM data during transformation, and then loading it to an XML message or file.

More and more database vendors allow you to store XML in one column. The field is usually a varchar, long, or clob. The software's XML handling capability also supports reading from and writing to such fields. The software provides four functions to support extracting from and loading to columns:

- `extract_from_xml`
- `load_to_xml`
- `long_to_varchar`
- `varchar_to_long`

The `extract_from_xml` function gets the XML content stored in a single column and builds the corresponding NRDM structure so that the software can transform it. This function takes varchar data only.

To enable extracting and parsing for columns, data from long and clob columns must be converted to varchar before it can be transformed by the software.

- The software converts a clob data type input to varchar if you select the *Import unsupported data types as VARCHAR of size* option when you create a database datastore connection in the Datastore Editor.
- If your source uses a long data type, use the long_to_varchar function to convert data to varchar.

i Note

The software limits the size of the XML supported with these methods to 100K due to the current limitation of its varchar data type. There are plans to lift this restriction in the future.

The function load_to_xml generates XML from a given NRDM structure in the software, then loads the generated XML to a varchar column. If you want a job to convert the output to a long column, use the varchar_to_long function, which takes the output of the load_to_xml function as input.

20.4.1 Sample scenarios

The following scenarios describe how to use functions to extract XML data from a source column and load it into a target column.

Related Information

[Extracting XML data from a column into the software \[page 685\]](#)

[Loading XML data into a column of the data type long \[page 687\]](#)

[Extract data quality XML strings using extract_from_xml function \[page 688\]](#)

20.4.1.1 Extracting XML data from a column into the software

This scenario uses long_to_varchar and extract_from_xml functions to extract XML data from a column with data type long.

Perform the following prerequisite steps before beginning the main steps:

1. Import an Oracle table that contains a column named *Content*, the data type long, and which contains XML data for a purchase order.
2. Import the XML Schema *PO.xsd*, that provides the format for the XML data, into the repository.
3. Create a project, a job, and a data flow for your design.
4. Open the data flow and drop the source table with the column named *content* in the data flow.

To extract XML data from a column into the software, follow the prerequisite steps above, and then follow the steps below:

1. Create a query with an output column of data type varchar, and make sure that its size is big enough to hold the XML data.

2. Name this output column *content*.
3. In the Map section of the query editor, open the Function Wizard, select the *Conversion* function type, then select the *long_to_varchar* function and configure it by entering its parameters.

```
long_to_varchar(content, 4000)
```

The second parameter in this function (4000 in the example above) is the maximum size of the XML data stored in the table column. Use this parameter with caution. If the size is not big enough to hold the maximum XML data for the column, the software will truncate the data and cause a runtime error. Conversely, do not enter a number that is too big, which would waste computer memory at runtime.

4. In the query editor, map the source table column to a new output column.
5. Create a second query that uses the function *extract_from_xml* to extract the XML data.

To invoke the function *extract_from_xml*:

- a. Right-click the current context in the query.
- b. Choose *New Function Call*.
- c. When the Function Wizard opens, select *Conversion* and *extract_from_xml*.

i Note

You can only use the *extract_from_xml* function in a new function call. Otherwise, this function is not displayed in the function wizard.

6. Enter values for the input parameters and click *Next*.

Input parameters

Parameter	Description
<i>XML column name</i>	Enter <i>content</i> , which is the output column in the previous query that holds the XML data.
<i>DTD or XML Schema name</i>	Enter the name of the purchase order schema (in this case <i>PO</i>).
<i>Enable validation</i>	Enter 1 if you want the software to validate the XML with the specified Schema. Enter 0 if you do not.

7. For the function, select a column or columns that you want to use on output.

Imagine that this purchase order schema has five top-level elements: *orderDate*, *shipTo*, *billTo*, *comment*, and *items*. You can select any number of the top-level columns from an XML schema, which include either scalar or nested relational data model (NRDM) column data. The return type of the column is defined in the schema. If the function fails due to an error when trying to produce the XML output, the software returns NULL for scalar columns and empty nested tables for NRDM columns. The *extract_from_xml* function also adds two columns:

- *AL_ERROR_NUM* — returns error codes: 0 for success and a non-zero integer for failures
- *AL_ERROR_MSG* — returns an error message if *AL_ERROR_NUM* is not 0. Returns NULL if *AL_ERROR_NUM* is 0

Choose one or more of these columns as the appropriate output for the *extract_from_xml* function.

8. Click *Finish*.

The software generates the function call in the current context and populates the output schema of the query with the output columns you specified.

With the data converted into the NRDM structure, you are ready to do appropriate transformation operations on it.

For example, if you want to load the NRDM structure to a target XML file, create an XML file target and connect the second query to it.

i Note

If you find that you want to modify the function call, right-click the function call in the second query and choose [Modify Function Call](#).

In this example, to extract XML data from a column of data type long, we created two queries: the first query to convert the data using the `long_to_varchar` function and the second query to add the `extract_from_xml` function.

Alternatively, you can use just one query by entering the function expression `long_to_varchar` directly into the first parameter of the function `extract_from_xml`. The first parameter of the function `extract_from_xml` can take a column of data type varchar or an expression that returns data of type varchar.

If the data type of the source column is not long but varchar, do not include the function `long_to_varchar` in your data flow.

20.4.1.2 Loading XML data into a column of the data type long

This scenario uses the `load_to_xml` function and the `varchar_to_long` function to convert an NRDM structure to scalar data of the varchar type in an XML format and load it to a column of the data type long.

In this example, you want to convert an NRDM structure for a purchase order to XML data using the function `load_to_xml`, and then load the data to an Oracle table column called [content](#), which is of the long data type. Because the function `load_to_xml` returns a value of varchar data type, you use the function `varchar_to_long` to convert the value of varchar data type to a value of the data type long.

1. Create a query and connect a previous query or source (that has the NRDM structure of a purchase order) to it. In this query, create an output column of the data type varchar called [content](#). Make sure the size of the column is big enough to hold the XML data.
2. From the [Mapping](#) area open the function wizard, click the category [Conversion Functions](#), and then select the function [load_to_xml](#).
3. Click [Next](#).
4. Enter values for the input parameters.

The function `load_to_xml` has seven parameters.

5. Click [Finish](#).

In the mapping area of the Query window, notice the function expression:

```
load_to_xml(PO, 'PO', 1, '<?xml version="1.0" encoding = "UTF-8" ?>', NULL, 1, 4000)
```

In this example, this function converts the NRDM structure of purchase order PO to XML data and assigns the value to output column content.

6. Create another query with output columns matching the columns of the target table.
 - a. Assume the column is called *content* and it is of the data type long.
 - b. Open the function wizard from the mapping section of the query and select the *Conversion Functions* category
 - c. Use the function `varchar_to_long` to map the input column *content* to the output column *content*.

The function `varchar_to_long` takes only one input parameter.
 - d. Enter a value for the input parameter.

```
varchar_to_long(content)
```

7. Connect this query to a database target.

Like the example using the `extract_from_xml` function, in this example, you used two queries. You used the first query to convert an NRDM structure to XML data and to assign the value to a column of varchar data type. You used the second query to convert the varchar data type to long.

You can use just one query if you use the two functions in one expression:

```
varchar_to_long( load_to_xml(PO, 'PO', 1, '<?xml version="1.0" encoding =  
"UTF-8" ?>', NULL, 1, 4000) )
```

If the data type of the column in the target database table that stores the XML data is varchar, there is no need for `varchar_to_long` in the transformation.

Related Information

[Reference Guide: Functions and Procedures \[page 773\]](#)

20.4.1.3 Extract data quality XML strings using `extract_from_xml` function

This scenario uses the `extract_from_xml` function to extract XML data from the Geocoder, Global Suggestion Lists, Global Address Cleanse, and USA Regulatory Address Cleanse transforms.

The Geocoder transform, Global Suggestion Lists transform, and the suggestion list functionality in the Global Address Cleanse and USA Regulatory Address Cleanse transforms can output a field that contains an XML string. The transforms output the following fields that can contain XML.

Transform	XML output field	Output field description
Geocoder	Result_List	Contains an XML output string when multiple records are returned for a search. The content depends on the available data.

Transform	XML output field	Output field description
Global Address Cleanse	Suggestion_List	Contains an XML output string that includes all of the suggestion list component field values specified in the transform options.
Global Suggestion List		
USA Regulatory Address Cleanse		
		To output these fields as XML, you must choose XML as the output style in the transform options.

To use the data contained within the XML strings (for example, in a web application that uses the job published as a web service), you must extract the data. There are two methods that you can use to extract the data:

Methods to extract data

Method	Description
Insert a Query transform using the <code>extract_from_xml</code> function	With this method, you insert a Query transform into the data flow after the Geocoder, Global Suggestion Lists, Global Address Cleanse, or USA Regulatory Address Cleanse transform. Then you use the <code>extract_from_xml</code> function to parse the nested output data. This method is considered a best practice, because it provides parsed output data that is easily available to an integrator.
Develop a simple data flow that does not unnest the nested data	With this method, you simply output the output field that contains the XML string without unnesting the nested data. This method allows the application developer, or integrator, to dynamically select the output components in the final output schema before exposing it as a web service. The application developer must work closely with the data flow designer to understand the data flow behind a real-time web service. The application developer must understand the transform options and specify what to return from the return address suggestion list, and then unnest the XML output string to generate discrete address elements.

Related Information

[Reference Guide: Functions and procedures, Descriptions of built-in functions, `extract\_from\_xml`](#)
[Overview of data quality \[page 347\]](#)

20.4.1.3.1 Extracting data quality XML strings using `extract_from_xml` function

This scenario uses the `extract_from_xml` function to extract XML strings.

1. Create an XSD file for the output.
2. In the [Format](#) tab of the Local Object Library, create an XML Schema for your output XSD.
3. In the [Format](#) tab of the Local Object Library, create an XML Schema for the `gac_suggestion_list.xsd`, `global_suggestion_list.xsd`, `urac_suggestion_list.xsd`, or `result_list.xsd`.
4. In the data flow, include the following field in the [Schema Out](#) of the transform:

Transform	Field
○ Global Address Cleanse ○ Global Suggestion Lists ○ USA Regulatory Address Cleanse	Suggestion_List
Geocoder	Result_List

5. Add a Query transform after the Global Address Cleanse, Global Suggestion Lists, USA Regulatory Address Cleanse, or Geocoder transform. Complete it as follows.
 - a. Pass through all fields except the Suggestion_List or Result_List field from the Schema In to the Schema Out. To do this, drag fields directly from the input schema to the output schema.
 - b. In the [Schema Out](#), right-click the Query node and select [New Output Schema](#). Enter Suggestion_List or Result_List as the schema name (or whatever the field name is in your output XSD).
 - c. In the [Schema Out](#), right-click the Suggestion_List or Result_List field and select [Make Current](#).
 - d. In the [Schema Out](#), right-click the Suggestion_List or Result_List list field and select [New Function Call](#).
 - e. Select [Conversion Functions](#) from the [Function categories](#) column and [extract_from_xml](#) from the [Function name](#) column and click [Next](#).
 - f. In the [Define Input Parameter\(s\)](#) window, enter the following information and click [Next](#).

Define Input Parameters options

Option	Description
XML field name	Select the Suggestion_List or Result_List field from the upstream transform.
DTD or Schema name	Select the XML Schema that you created for the <code>gac_suggestion_list.xsd</code> , <code>urac_suggestion_list.xsd</code> , or <code>result_list.xsd</code> .
Enable validation	Enter 1 to enable validation.

- g. Select LIST or RECORD from the left parameter list and click the right arrow button to add it to the Selected output parameters list.
- h. Click [Finish](#).

The [Schema Out](#) includes the suggestion list/result list fields within the Suggestion_List or Result_List field.

6. Include the XML Schema for your output XML following the Query. Open the XML Schema to validate that the fields are the same in both the [Schema In](#) and the [Schema Out](#).

7. If you are extracting data from a Global Address Cleanse, Global Suggestion Lists, or USA Regulatory Address Cleanse transform, and have chosen to output only a subset of the available suggestion list output fields in the *Options* tab, insert a second Query transform to specify the fields that you want to output. This allows you to select the output components in the final output schema before it is exposed as a web service.

Related Information

[Reference Guide: Functions and Procedures, Descriptions of built-in functions, extract_from_xml](#)
[Overview of data quality \[page 347\]](#)

20.5 JSON extraction

In addition to extracting JSON message and file data, representing it as NRDM data during transformation, and then loading it to an JSON message or file, you can also use the software to extract JSON data stored in a source table or flat file column.

The software provides the `extract_from_json` function to support extracting from columns.

The `extract_from_json` function gets the JSON content stored in a single column and builds the corresponding NRDM structure so that the software can transform it. This function takes varchar data only.

To enable extracting and parsing for columns, data from long and clob columns must be converted to varchar before it can be transformed by the software.

- The software converts a clob data type input to varchar if you select the *Import unsupported data types as VARCHAR of size* option when you create a database datastore connection in the Datastore Editor.
- If your source uses a long data type, use the `long_to_varchar` function to convert data to varchar.

20.5.1 Extracting data from JSON string using `extract_from_json` function

1. Create a schema file or JSON data file for the output.
2. In the *Format* tab of the Local Object Library, click **New** > *JSON Schema* to create a Schema for your output using above schema file or JSON data file.
3. In the data flow, in the schema out, right-click a column and select *New Function Call*.
4. In the *Conversion Functions* category, select *extract_from_json* and click *Next*.
5. In the Define Input Parameter(s) window, enter the following information and click *Next*:

Option	Description
JSON field name	Select the Suggestion_List or Result_List field from the upstream transform.
Schema name	Select the Schema that you created
Enable validation	Enter 1 to enable validation.

6. Select *LIST* or *RECORD* from the left parameter list and click the right arrow button to add it to the selected output parameters list.
7. Click *Finish*.
The Schema Out includes the suggestion list/result list fields within the Suggestion_List or Result_List field.

21 Embedded Data Flows

An embedded data flow is a data flow that is called from inside another data flow.

The software provides easy-to-use options to create embedded data flows.

21.1 Overview of embedded data flows

Data passes into or out of the embedded data flow from the parent flow through a single source or target.

An embedded data flow can contain any number of sources or targets, but only one input or one output can pass data to or from the parent data flow.

You can create the following types of embedded data flows:

Type	Use when you want to...
One input	Add an embedded data flow at the end of a data flow.
One output	Add an embedded data flow at the beginning of a data flow.
No input or output	Replicate an existing data flow.

An embedded data flow is a design aid that has no effect on job execution. When the software executes the parent data flow, it expands any embedded data flows, optimizes the parent data flow, and then executes it.

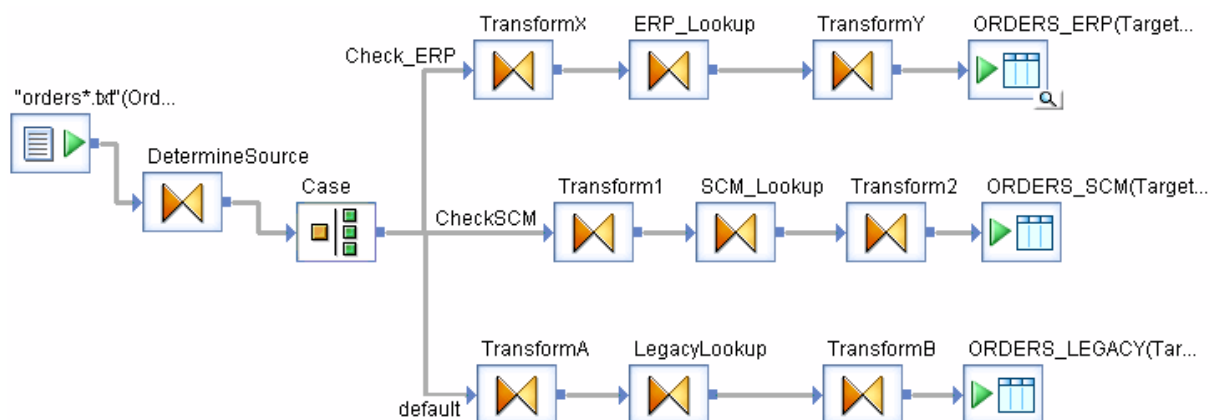
Use embedded data flows to:

- Simplify data flow display. Group sections of a data flow in embedded data flows to allow clearer layout and documentation.
- Reuse data flow logic. Save logical sections of a data flow so you can use the exact logic in other data flows, or provide an easy way to replicate the logic and modify it for other flows.
- Debug data flow logic. Replicate sections of a data flow as embedded data flows so you can execute them independently.

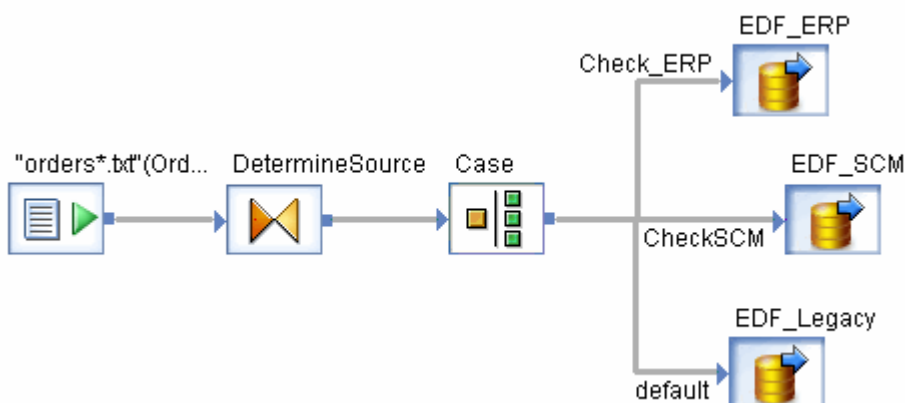
21.2 Embedded data flow examples

Examples of when and how to use embedded data flows.

In this example, a data flow uses a single source to load three different target systems. The Case transform sends each row from the source to different transforms that process it to get a unique target output.



You can simplify the parent data flow by using embedded data flows for the three different cases.



21.3 Creating embedded data flows

You can create a new embedded data flow or embed an existing data flow.

To create embedded data flows, you can use one of two methods:

- Select objects within a data flow, right-click, and select [Make Embedded Data Flow](#).
- Drag a complete and fully validated data flow from the object library into an open data flow in the workspace. Then:
 - Open the data flow you just added.
 - Right-click one object you want to use as an input or as an output port and select [Make Port](#) for that object.
 The software marks the object you select as the connection point for this embedded data flow.

i Note

You can specify only one port, which means that the embedded data flow can appear only at the beginning or at the end of the parent data flow.

Data Services ignores some physical files that are required for sources and targets with assigned ports.

- When you use an embedded data flow, data will flow directly from the caller to the transform(s) next to the port source.
- When you use a data flow directly, Data Services uses the physical files in sources and targets, but ignores the ports.

21.3.1 Using the Make Embedded Data Flow option

Use this option to create a new embedded data flow.

1. Select objects from an open data flow using one of the following methods:
 - Click the white space and drag the rectangle around the objects.
 - CTRL-click each object.

Ensure that the set of objects you select are:

- All connected to each other.
- Connected to other objects according to the type of embedded data flow you want to create such as one input, one output, or no input or output.

2. Right-click and select *Make Embedded Data Flow*.

The Create Embedded Data Flow window opens, with the embedded data flow connected to the parent by one input object.

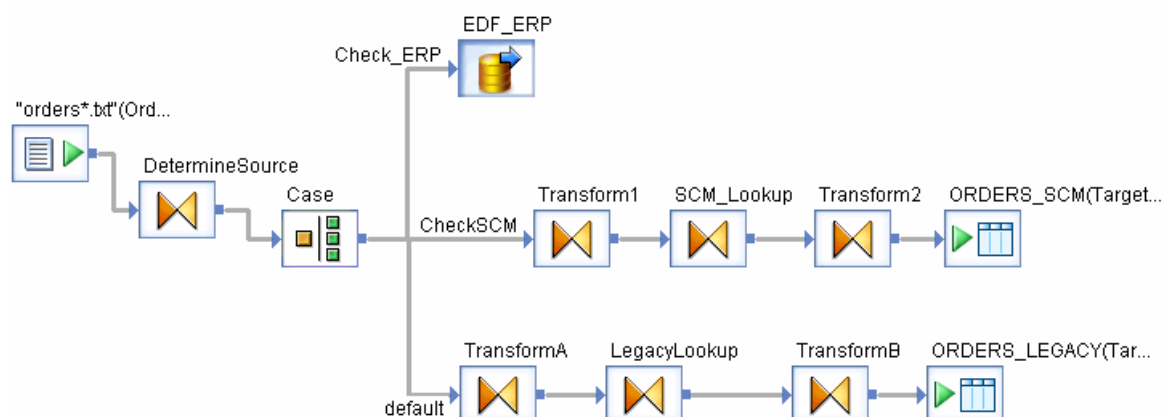
3. Name the embedded data flow using the convention EDF_<EDFName> for example EDF_ERP.

If you deselect the Replace objects in original data flow box, the software will not make a change in the original data flow. The software saves the new embedded data flow object to the repository and displays it in the object library under the Data Flows tab.

You can use an embedded data flow created without replacement as a stand-alone data flow for troubleshooting.

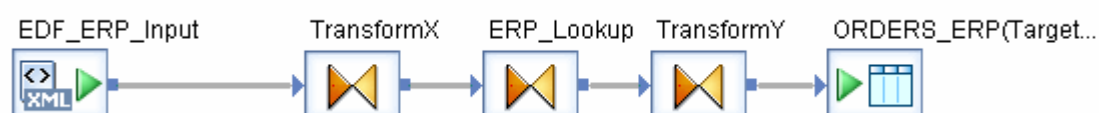
If Replace objects in original data flow is selected, the original data flow becomes a parent data flow, which has a call to the new embedded data flow.

4. Click *OK*.



The embedded data flow appears in the new parent data flow.

- Click the name of the embedded data flow to open it.



- Notice that the software created a new object, `EDF_ERP_Input`, which is the input port that connects this embedded data flow to the parent data flow.

When you use the [Make Embedded Data flow](#) option, the software automatically creates an input or output object based on the object that is connected to the embedded data flow when it is created.

For example, if an embedded data flow has an output connection, the embedded data flow will include a target XML file object labeled `<EDFName>_Output`.

The naming conventions for each embedded data flow type are:

Type	Naming conventions
One input	<code><EDFName>_Input</code>
One output	<code><EDFName>_Output</code>
No input or output	The software creates an embedded data flow without an input or output object.

21.3.2 Creating embedded data flows from existing flows

You can call an existing data flow from inside another data flow.

You will need to put the data flow inside the parent data flow, then mark which source or target to use to pass data between the parent and the embedded data flows.

1. Drag an existing valid data flow from the object library into a data flow that is open in the workspace.
2. Consider renaming the flow using the `EDF_<EDFName>` naming convention.

The embedded data flow appears without any arrowheads (ports) in the workspace.

3. Open the embedded data flow.
4. Right-click a source or target object (file or table) and select *Make Port*.

i Note

Ensure that you specify only one input or output port.

Like a normal data flow, different types of embedded data flow ports are indicated by directional markings on the embedded data flow icon.

21.3.3 Using embedded data flows

When you create and configure an embedded data flow using the *Make Embedded Data Flow* option, the software creates new input or output XML file and saves the schema in the repository as an XML Schema.

You can reuse an embedded data flow by dragging it from the *Data Flow* tab of the object library into other data flows. To save mapping time, you might want to use the Update Schema option or the Match Schema option.

The following example scenario uses both options:

- Create data flow 1.
- Select objects in data flow 1, and create embedded data flow 1 so that parent data flow 1 calls embedded data flow 1.
- Create data flow 2 and data flow 3 and add embedded data flow 1 to both of them.
- Go back to data flow 1. Change the schema of the object preceding embedded data flow 1 and use the Update Schema option with embedded data flow 1. It updates the schema of embedded data flow 1 in the repository.
- Now the schemas in data flow 2 and data flow 3 that are feeding into embedded data flow 1 will be different from the schema the embedded data flow expects.
- Use the Match Schema option for embedded data flow 1 in both data flow 2 and data flow 3 to resolve the mismatches at runtime. The Match Schema option only affects settings in the current data flow.

The following sections describe the use of the Update Schema and Match Schema options in more detail.

21.3.3.1 Updating Schemas

The software provides an option to update an input schema of an embedded data flow.

This option updates the schema of an embedded data flow's input object with the schema of the preceding object in the parent data flow. All occurrences of the embedded data flow update when you use this option.

1. Open the embedded data flow's parent data flow.
2. Right-click the embedded data flow object and select *Update Schema*.

21.3.3.2 Matching data between parent and embedded data flow

The schema of an embedded data flow's input object can match the schema of the preceding object in the parent data flow by name or position. A match by position is the default.

21.3.3.2.1 Specifying how schemas should be matched

1. Open the embedded data flow's parent data flow.
2. Right-click the embedded data flow object and select [Match SchemaBy Name](#) or [Match SchemaBy Position](#).

The Match Schema option only affects settings for the current data flow.

Data Services also allows the schema of the preceding object in the parent data flow to have more or fewer columns than the embedded data flow. The embedded data flow ignores additional columns and reads missing columns as NULL.

Columns in both schemas must have identical or convertible data types. See the section on "Type conversion" in the *Reference Guide* for more information.

21.3.3.3 Deleting embedded data flow objects

You can delete embedded data flow ports, or remove entire embedded data flows.

21.3.3.3.1 Removing a port

Right-click the input or output object within the embedded data flow and deselect [Make Port](#). Data Services removes the connection to the parent object.

i Note

You cannot remove a port simply by deleting the connection in the parent flow.

21.3.3.3.2 Removing an embedded data flow

Select it from the open parent data flow and choose [Delete](#) from the right-click menu or edit menu.

If you delete embedded data flows from the object library, the embedded data flow icon appears with a red circle-slash flag in the parent data flow.

EDF_Example



Delete these defunct embedded data flow objects from the parent data flows.

21.3.4 Separately testing an embedded data flow

Embedded data flows can be tested by running them separately as regular data flows.

1. Specify an XML file for the input port or output port.

When you use the Make Embedded Data Flow option, an input or output XML file object is created and then (optional) connected to the preceding or succeeding object in the parent data flow. To test the XML file without a parent data flow, click the name of the XML file to open its source or target editor to specify a file name.

2. Put the embedded data flow into a job.
3. Run the job.

You can also use the following features to test embedded data flows:

- View Data to sample data passed into an embedded data flow.
- Auditing statistics about the data read from sources, transformed, and loaded into targets, and rules about the audit statistics to verify the expected data is processed.

Related Information

[Reference Guide: Objects, Descriptions of objects, XML file Design and Debug \[page 815\]](#)

21.3.5 Troubleshooting embedded data flows

The following situations produce errors:

- Both an input port and output port are specified in an embedded data flow.
- Trapped defunct data flows.
- Deleted connection to the parent data flow while the *Make Port* option, in the embedded data flow, remains selected.
- Transforms with splitters (such as the Case transform) specified as the output port object because a splitter produces multiple outputs, and embedded data flows can only have one.
- Variables and parameters declared in the embedded data flow that are not also declared in the parent data flow.
- Embedding the same data flow at any level within itself.

You can however have unlimited embedding levels. For example, DF1 data flow calls EDF1 embedded data flow which calls EDF2.

Related Information

[Removing an embedded data flow \[page 698\]](#)

[Removing a port \[page 698\]](#)

22 Variables and Parameters

Increase the flexibility and reusability of jobs, work flows, and data flows by using variables and parameters.

Variables are symbolic placeholders for values. Assign a data type to a variable based on the value that will populate the variable. Data types can be any that Data Services supports such as an integer, decimal, date, or text string.

Parameters are expressions that pass to a work flow or data flow when they are called in a job. Designate a parameter as an input, output, or input and output type.

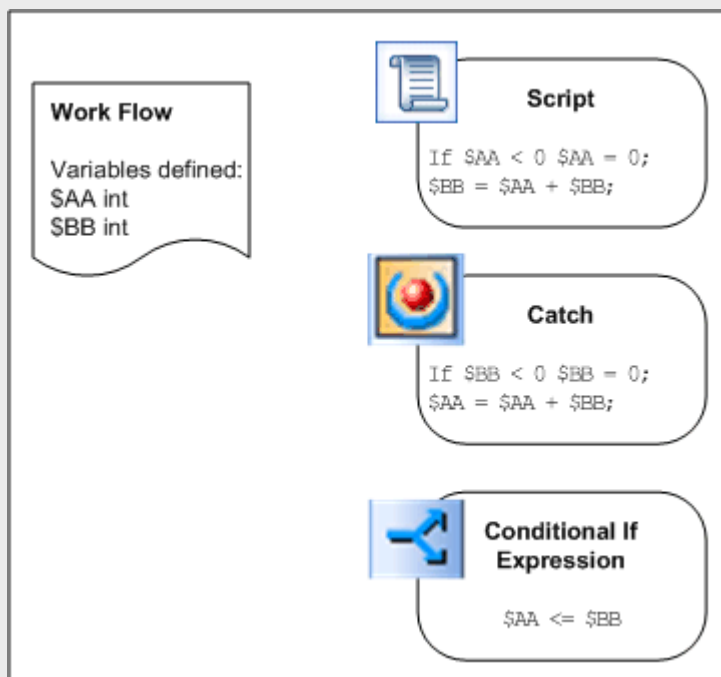
You can use variables in expressions to facilitate decision-making or data manipulation (using arithmetic or character substitution). For example, a variable can be used in a `LOOP` or `IF` statement to base an action on the value in the variable.

❖ Example

```
If $amount_owed > 0 print('$invoice.doc');
```

If you define local variables in a job or work flow, you also typically include scripts, catch, or conditional processes.

❖ Example



There are two types of variables, each with its own restrictions.

Variable type	Restrictions
Local	• Restricted to the object in which they are created, such as a job or work flow. • Passed as parameters to child objects. For example, if variable is in data flow, pass value to work flow.
Global	• Restricted to the job in which they are created. • Does not require parameters to be passed to work flows and data flows.

i Note
 If you have work flows that are running in parallel, Data Services does not assign global variables.

Create local variables, parameters, and global variables using the [Variables and Parameters](#) dialog box in Designer.

Set values for local or global variables in script objects. You can also set global variable values using external job, execution, or schedule properties.

You can use variables as file names for the following source and target types:

- Flat file sources and targets
- XML file sources and targets
- XML message targets (executed in the Designer in test mode)
- IDoc file sources and targets (in an SAP application environment)
- IDoc message sources and targets (SAP application environment)

[Viewing variables and parameters \[page 703\]](#)

Use the [Variables and Parameters](#) dialog box to view, edit, and create local and global variables and parameters for selected objects in Designer.

[Variables and Parameters dialog box \[page 703\]](#)

The [Variables and Parameters](#) dialog box stays on top of whatever you have opened in the project area and workspace area of the Designer.

[Definitions and Calls tabs \[page 704\]](#)

The [Variables and Parameters](#) dialog box contains the [Definitions](#) tab and the [Calls](#) tab.

[Object types and uses for variables and parameters \[page 705\]](#)

Use variables and parameters based on the object type and whether the variable is local or global.

[Accessing View Where Used for variables and parameters \[page 706\]](#)

Find out what other objects use the variables and parameters by using the [View Where Used](#) option.

[Use local variables and parameters \[page 706\]](#)

Local variables and parameters are restricted to the object in which they are created like a job or a work flow.

[Use global variables \[page 711\]](#)

Global variables are restricted to the job in which they are created.

[Local and global variable rules \[page 717\]](#)

There are rules to follow when you set values for local and global variables.

[Environment variables \[page 717\]](#)

[Set file names at run-time using variables \[page 718\]](#)

You can set file names at runtime by specifying a variable as the file name.

Related Information

[Management Console Guide: Administrator, Support for Web Services](#)

22.1 Viewing variables and parameters

Use the [Variables and Parameters](#) dialog box to view, edit, and create local and global variables and parameters for selected objects in Designer.

Open a project in the project area, or select any object in the project.

1. Select  [Tools](#) .

The [Variables and Parameters](#) dialog box opens. The name of the selected object appears in the [Context](#) label under the header of the dialog box. All variables and parameters assigned to the object appear in the dialog box.

2. Click another object in the project area or the workspace area.

The [Context](#) label changes to reflect the name of the selected object. The variables and parameters change based on the object you select.

22.2 Variables and Parameters dialog box

The [Variables and Parameters](#) dialog box stays on top of whatever you have opened in the project area and workspace area of the Designer.

Drag the dialog box to a different position if it is in the way, or dock it just like you do for the project area and local object library. When you click other applicable objects, Data Services changes the [Variables and Parameters](#) dialog box [Context](#) label at the top to display the name of the selected or open object. Data Services updates the list of variables and parameters for the selected object. If the object isn't applicable for variables and parameters, or if it doesn't have any variables and parameters set, the dialog content does not change, or it is blank.

There are two tabs at the bottom of the dialog box: [Definitions](#) and [Calls](#).

- [Definitions](#): View and create variables and parameters.
- [Calls](#): View the argument value of each variable and parameter that is defined for all objects in the [Definitions](#) tab of the object.

Additionally, right-click the name of any variable or parameter in either tab and select [View Where Used](#) from the context menu to see information about other objects that use the variable or parameter.

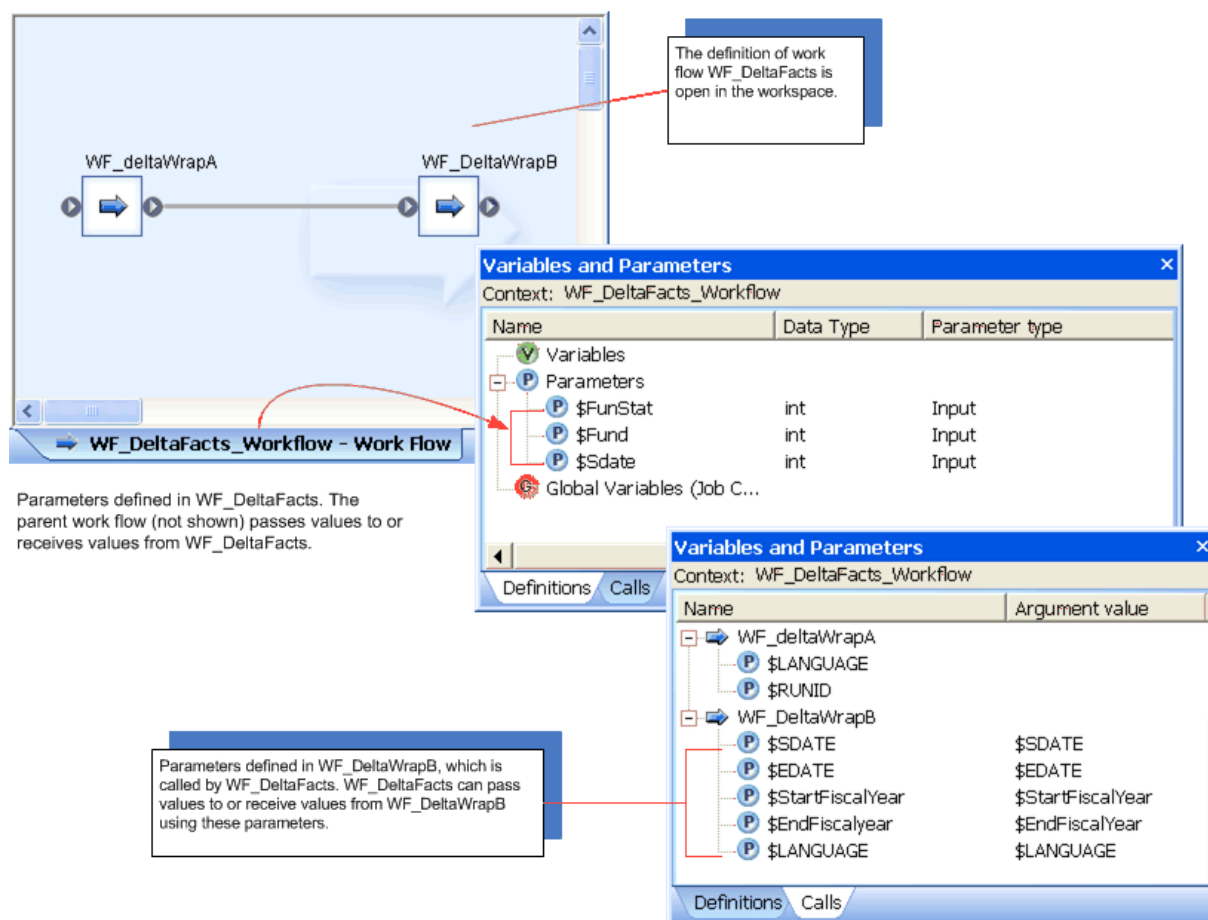
22.3 Definitions and Calls tabs

The [Variables and Parameters](#) dialog box contains the [Definitions](#) tab and the [Calls](#) tab.

Tab	Purpose
Definitions	View the applicable parameters and variables for the selected object. For variables, see the data type and value. For parameters, see the data type and parameter type, which indicates if the parameter is for input, output, or both. Also insert new variables and parameters based on the selected object type: • Set local variables and parameters at the work flow and data flow levels. • Set global variables at the job level.
Calls	View the name of each child object and parameter that is defined for the child object in the selected parent object. Enter values for each parameter as follows: • Input parameter types: Values can be constants, variables, or another parameter. • Output or input/output parameter types: Values can be variables or parameters. Additionally, values in the Calls tab must use: • The same data type as the variable when the values are inside an input or input/output parameter type. • Compatible with the variable data type when the values are placed inside an output parameter type. • Adhere to scripting language rules and syntax.

Example

The following illustration shows the contents of the [Definitions](#) and [Calls](#) tabs in the [Variables and Parameters](#) window for the open work flow named `WF_DeltaFacts`. Notice the work flow name displays in the [Context](#) label at the top of the [Variables and Parameters](#) window.



22.4 Object types and uses for variables and parameters

Use variables and parameters based on the object type and whether the variable is local or global.

The following table lists the types of variables and parameters you can create based on the object type and how you may use them.

Object type	Applicable variables and parameters	Used by
Job	Local variables	A script or conditional in the job.
	Global variables	Any object in the job.
Work flow	Local variables	Current work flow or passed down to other work flows or data flows using a parameter.

Object type	Applicable variables and parameters	Used by
	Parameters	Parent objects that pass local parameters. Work flows may also return parameters to parent objects.
Data flow	Parameters	A WHERE clause, column mapping, or a function in a data flow. Data flows cannot return output values.

22.5 Accessing View Where Used for variables and parameters

Find out what other objects use the variables and parameters by using the [View Where Used](#) option.

In Designer, open the project and select the project or child object that contains the variable or parameter for which you want to view where else it is used.

1. Select **Tools > Variables**.

The [Variables and Parameters](#) dialog box opens.

2. In either the [Definitions](#) or [Calls](#) tab, right-click the name of the applicable parameter or variable, and select [View Where Used](#) from the context menu.

A [Search](#) dialog box opens showing a list of all objects that use the variable.

i Note

You do not have to change any settings in the [Search](#) dialog. The software automatically populates search criteria to find the information for the selected variable.

Search for use information for a different variable or parameter without closing either the [Search](#) dialog box or the [Variables and Parameters](#) dialog box. Select a different variable or parameter or a different object from the project area and perform the same steps to obtain use information for a different variable or parameter.

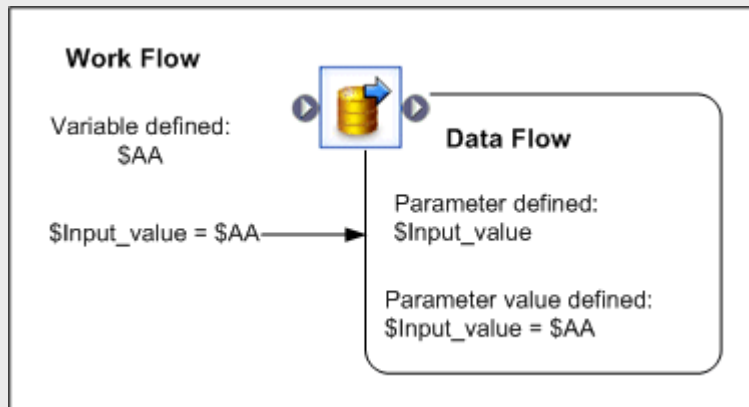
22.6 Use local variables and parameters

Local variables and parameters are restricted to the object in which they are created like a job or a work flow.

To pass a local variable to another object, define the local variable, create a parameter in the calling object, and map the parameter to the local variable by entering a parameter value.

❖ Example

To use a local variable inside a data flow, define the variable in a parent work flow and then pass the value of the variable as a parameter of the data flow.



22.6.1 Parameters

Define parameters to pass their values into and out of work flows, and to pass values into data flows.

When you create a parameter, you assign a parameter type: Input, output, or input/output. You also assign the data type. Any object can use the value passed by the parameter when it is called by an associated work flow or data flow.

i Note

You can also create local variables and parameters for use in custom functions.

Related Information

[Reference Guide: Functions and Procedures, Custom functions \[page 778\]](#)

22.6.2 Adding a parameter to a work flow or data flow

After you create a parameter, add it to the related work flow or data flow to pass a value into the flow, and—or pass a value to the output object in a flow.

1. Click the name of the work flow or data flow in the project area or workspace in Designer.
2. Open the *Variables and Parameters* dialog box by selecting **Tools** > *Variables*.

Ensure that the *Context* label in the *Variables and Parameters* dialog box contains the name of the work flow or data flow you selected.

3. In the *Definitions* tab, right-click *Parameters* and select *Insert*.

Optional. If there are current parameters listed, select the parameter closest to where you want to insert the new parameter, right-click, and select *Insert Before* or *Insert After* as applicable.

A new parameter appears with a base name such as `$NewParameter0`.

4. Double-click the new parameter to open the *Parameter Properties* dialog box.
5. Enter a name for the new parameter.

The name can include alphanumeric characters or underscores (`_`), but cannot contain blank spaces. Always begin the name with a dollar sign (`$`).

→ Tip

In addition to the dollar sign, prefix the parameter name with the initials “LP” for local parameter, followed by the parameter type: “O” for output, “I” for input, and “IO” for input/output. For example, for a local parameter that is for input, prefix the parameter name with “\$LP_I_”. The prefix helps you to more easily identify the parameter and purpose.

6. Select the applicable data type from the *Data type* dropdown list.

Select from a list of all supported data types.

7. Select the applicable type from the *Parameter type* dropdown list. Options include *Input*, *Output*, and *Input/Output*.

If you select *Input* or *Input/Output*, ensure that the data type you choose is the same as the value you use with the parameter. If you select *Output*, ensure that the data type of the parameter is compatible with the value you use with the parameter. Additionally, the value can be an expression, variable, or another parameter based on the parameter type.

8. Optional. To place your parameters in a specific order, right-click the parameter and select *Move Up* or *Move Down*.
9. To close the *Variables and Parameters* dialog box, click the “X” icon in the upper right.

22.6.3 Pass values into data flows

To control the data transformed in a data flow, use a value passed as a parameter into the data flow.

❖ Example

The data flow **DF_PartFlow** processes daily inventory values. It can process all of the part numbers in use or a range of part numbers based on external requirements such as the range of numbers processed most recently.

If the work flow that calls **DF_PartFlow** records the range of numbers processed, it can pass the end value of the range `$EndRange` as a parameter to the data flow to indicate the start value of the range to process next.

Data Services calculates a new end value based on a variable named `$SizeOfSet`, that stores the number of parts to process each time. Data Services passes the value of `$SizeOfSet` to the data flow as the end value. A query transform in the data flow uses the parameters passed in to filter the part numbers extracted from the source.

Work Flow

Variables defined:
\$EndRange
\$SizeOfSet

Parameter values defined:
\$StartID = \$EndRange = 1
\$EndID = \$EndRange + \$SizeOfSet



DF_PartFlow

Parameters defined:
\$StartID
\$EndID

Use the data flow multiple times in the same work flow or different work flows to perform the same task on different part number ranges. Specify different parameters for each call.

22.6.4 Setting the value of the parameter in the flow call

A parameter needs a value to pass based on the parameter settings.

Open the project that contains the calling job, work flow, or data flow for the applicable parameter in the project area. Select the calling object or open it in the workspace.

1. Open the *Variables and Parameters* dialog box by selecting **Tools > Variables**.
2. Open the *Calls* tab in the *Variables and Parameters* dialog box.

The *Calls* tab shows all the parameters that are called from the selected calling object.

3. Click the *Argument Value* cell next to the applicable parameter.
The cell turns into an editable text box.
4. Enter the expression in the cell.

The expression is what the object passes to the designated object in the work flow or data flow. If the parameter type is *Input*, then the value can be an expression, variable, or another parameter. For example, an expression could contain a constant such as 0, 3, or 'string1'.

If the parameter type is *Output* or *Input/Output*, then the value can be a variable or parameter.

Note

The value cannot be a constant because, by definition, the value of an output or input/output parameter is modifiable by any object within the flow.

To indicate special values, use the following syntax:

Value type	Special syntax
Variable	\$variable_name
String	'string '

22.6.5 Defining a local variable

Create and define a new local variable for a work flow or job using the *Variables and Parameters* dialog box.

1. In Designer, click the name of the job or work flow in the project area or workspace, or double-click one from the object library to open it in the workspace.
2. Click  **Tools** > **Variables** .
3. In the *Definitions* tab, right-click *Variables* (indicated by a green circle with a V icon) and select *Insert*.
A new variable appears in the list with the temporary name such as \$NewVariable0.
4. Optional. If there are existing parameters, select a variable near the location for the new variable, right-click and select *Insert before* or *Insert after* as applicable.
5. Double-click the new variable to open the *Parameter Properties* dialog box.
6. Enter a new name for the variable starting with a dollar sign (\$).

The name can include alphanumeric characters or underscores (_), but cannot contain blank spaces. Always begin the name with a dollar sign (\$).

→ Tip

Consider using a naming convention that helps you identify the variable. Include a prefix after the dollar sign that indicates “L” for local and “V” for variable. For example, \$LV_variable1.

7. Select the appropriate data type from the *Data type* dropdown list.
 8. Select the applicable type from the Parameter type dropdown list.
 9. Click *OK*.
- The Variables and Parameters dialog box appears showing your newly added variable.
10. Optional. To change the order of the new variable in the list, right-click the variable name and select *Move Up* or *Move Down*.

22.6.6 Replicating a local variable

Copy an existing local variable to create a variable based on an existing variable.

1. In Designer, click the name of the job or work flow in the project area or workspace, or double-click one from the object library.
2. Click  **Tools** > **Variables** .
3. With the *Definition* tab opened, right-click the name of the Variable to copy and select *Replicate*.
A properties window opens and displays a new variable name with the same properties as the replicated variable.
4. Type a new name for the variable, and make any other changes to the Data type as applicable.

The name can include alphanumeric characters or underscores (_), but cannot contain blank spaces. Always begin the name with a dollar sign (\$).

→ Tip

Consider using a naming convention that helps you identify the variable. Include a prefix after the dollar sign that indicates “L” for local and “V” for variable. For example, `$LV_variable1`.

5. Click *OK*.

The new variable appears in the list of variables in the *Definition* tab.

6. Optional. To move the variable to a new location in the list, right-click the name of the variable in the list and select *Move Up* or *Move Down*.

22.7 Use global variables

Global variables are restricted to the job in which they are created.

When you assign a name to a global variable, that name is exclusive to the job. You cannot reuse the same global variable or the name of an existing global variable in a different job.

Global variables provide you with maximum flexibility at runtime. For example, during production you can change values for default global variables at runtime from a job schedule or SOAP call without having to open a job in the Designer.

You can set values for global variables in script objects, or you can use external job, execution, or schedule properties.

Valid global variables include variables for tasks at the job level such as the following:

- A variable that indicates the job should be executed in recovery mode.
- A variable that indicates the date and time that a job should run.
- A variable that indicates the date and time that a job completed.

22.7.1 Defining a global variable

Define a global variable in the *Variables and Parameters* dialog box in the applicable job.

1. In Designer, click the name of the job in the project area or workspace, or double-click one from the object library.
2. Click **Tools** > *Variables*.

The *Variables and Parameters* dialog box appears. The *Context* label displays the job name, and the dialog box displays any currently defined variables for this job.

3. In the *Definition* tab, right-click *Global Variables* and select *Insert*.
4. Optional. If there are existing variables, select a variable near the location for the new parameter, right-click, and select *Insert Before* or *Insert After*.

The new variable appears in the list with a temporary name such as `$NewJobGlobalVariable0`.

5. Double-click the new variable to open the *Global Variable Properties* dialog box.

6. Enter a new name for the global variable.

The name can include alphanumeric characters or underscores (_), but cannot contain blank spaces. Always begin the name with a dollar sign (\$).

→ Tip

Consider using a naming convention that helps you identify the variable. Include a prefix after the dollar sign that indicates “G” for global and “V” for variable. For example, \$GV_variable1.

7. Select the appropriate data type from the *Data type* dropdown list.
8. To close the dialog box, click *OK*.

22.7.2 Replicating a global variable

Copy an existing global variable to create a variable based on an existing variable.

1. In Designer, click the name of the job in the project area or workspace, or double-click a job name from the object library.
2. Click **Tools** > *Variables*.

The *Variables and Parameters* dialog box appears. The dialog box displays any currently defined variables for this object.

3. In the *Definition* tab, right-click the name of the global variable to copy and select *Replicate*.

A properties window opens and displays a new variable name with the same properties as the replicated variable.

4. Type a new name for the variable, and make any other changes to the Data type as applicable.

The name can include alphanumeric characters or underscores (_), but cannot contain blank spaces. Always begin the name with a dollar sign (\$).

→ Tip

Consider using a naming convention that helps you identify the variable. Include a prefix after the dollar sign that indicates “G” for global and “V” for variable. For example, \$GV_variable1.

5. Click *OK*.

The new global variable appears in the list of global variables in the *Definition* tab.

6. Optional. To move the variable to a new location in the list, right-click the name of the global variable and select *Move Up* or *Move Down*.

22.7.3 Viewing global variables from the Properties dialog

While you can view global variables for a job using the *Variables and Parameters* dialog box, you can also view global variables from the *Properties* dialog box of the selected job.

Global variables, defined in a job, are visible to those objects relative to that job. A global variable defined in one job is not available for modification or viewing from another job.

1. Open the [Jobs](#) tab in the object library.
2. Right-click the applicable job and select [Properties](#).
3. Click the [Global Variable](#) tab.
4. Optional. Use the [Filter](#) box to filter the list of global variables based on specific characteristics.

22.7.4 Values for global variables

In addition to setting a global variable value inside a job using an initialization script, you can set and maintain global variable values outside a job as well.

Values set outside a job are processed the same way as values set inside a job, for example, in an initialization script. Both methods have their advantages.

When you have internal and external values set for a global variable in a job, the software automatically chooses the value that you set inside the job as the value to use.

There are some advantages to setting values outside a job. For example, you can open the [Properties](#) dialog box for a job to view global substitution values instead of opening up the job.

You set global variable values outside a job in the following ways:

- As a job property
- As an execution or schedule property

i Note

You don't have to assign a value to a global variable. However, the software reads a global variable without a value as NULL.

When you set values for global variables outside a job, you can view the values in the [Properties](#) and the [Execution Properties](#) dialog boxes in Designer and in the [Execution Options](#) and [Schedule](#) pages of the Administrator. Additionally, you can easily edit the values when you test or schedule the job.

i Note

You cannot pass global variables as command-line arguments for real-time jobs.

22.7.5 Setting a global variable value as a job property

Set a value for a global variable in the [Properties](#) dialog box for the associated job.

1. Right-click the applicable job in the object library or project area and select [Properties](#).
2. Open the [Global Variable](#) tab.

All global variables created in the job appear with the Type and Value columns.

3. Optional. When there are numerous variables associated with the job, filter the global variables by name by entering part of the name in the [Filter](#) box.

4. Double-click the cell under the Value column for the applicable variable and enter a valid value.
You can use any statement that you would use in a script with this option.
5. To set multiple global variable values, use the `Ctrl` button to select the variables, right-click, and select [Update Value](#). Enter the value in the Smart Editor and click [OK](#).
6. Click [OK](#).

The software saves the new values in the repository as job properties.

You can also view and edit the values in the [Execution Properties](#) dialog of the Designer and in the [Execution Options](#) and [Schedule](#) pages of the Administrator. Setting the values in these locations enables you to override job property values at run time.

Related Information

[Reference Guide: Scripting Language \[page 803\]](#)

22.7.6 Setting a global variable value as an execution property

1. Execute a job from the Designer, or execute or schedule a batch job from the Administrator.

i Note

For testing purposes, you can execute real-time jobs from the Designer in test mode. Make sure to set the execution properties for a real-time job.

2. View the global variables in the job and their default values (if available).
3. Edit values for global variables as desired.
4. If you are using the Designer, click [OK](#). If you are using the Administrator, click [Execute](#) or [Schedule](#).

The job runs using the values you enter. Values entered as execution properties are not saved. Values entered as schedule properties are saved but can only be accessed from within the Administrator.

22.7.7 Automatic ranking of global variable values in a job

When you set values for global variables, and you enter different values for the same global variable using different methods, the software uses automatic ranking to select the value to use for a job run.

The software selects the highest ranking value for use in the job. A value entered as a job property has the lowest rank. A value defined inside a job has the highest rank.

22.7.7.1 Execution property versus job property

If you set a global variable value using a job property, and you set a different global variable (for the same global variable) using an execution property, the execution property value overrides the job property value and becomes the default value for the current job run.

i Note

You cannot save execution property global variable values.

Example

Assume that a job named JOB_Test1 has three global variables declared:

- \$YEAR
- \$MONTH
- \$DAY

You set the variable "\$YEAR" as a job property with a value of "2003".

For the initial job run, you set variables values for \$MONTH and \$DAY as execution properties as follows:

- \$MONTH = JANUARY
- \$DAY = 31

The software executes a list of statements which includes default values for JOB_Test1:

```
$YEAR=2003;  
$MONTH='JANUARY';  
$DAY=31;
```

For the second job run, you set variables \$YEAR = "2002" and \$MONTH = "JANUARY" as execution properties but you don't set a value for \$DAY. The software executes the following list of statements:

```
$YEAR=2002;  
$MONTH='JANUARY';
```

Because you do not set a value for \$DAY during execution for the second job, the software reads it as NULL, and the \$YEAR value is 2002 and not 2003, because you set it as an execution property to "2002" for the second run.

22.7.7.2 Job property versus schedule property

If you set a global variable value for both a job property and a schedule property, the schedule property value overrides the job property value and becomes the external default value for the current job run.

The software saves the schedule property value in the repository. However, these values are only associated with a job schedule, not the job itself. Consequently, you must view and edit these values from within the Administrator.

22.7.7.3 Inside job versus external property

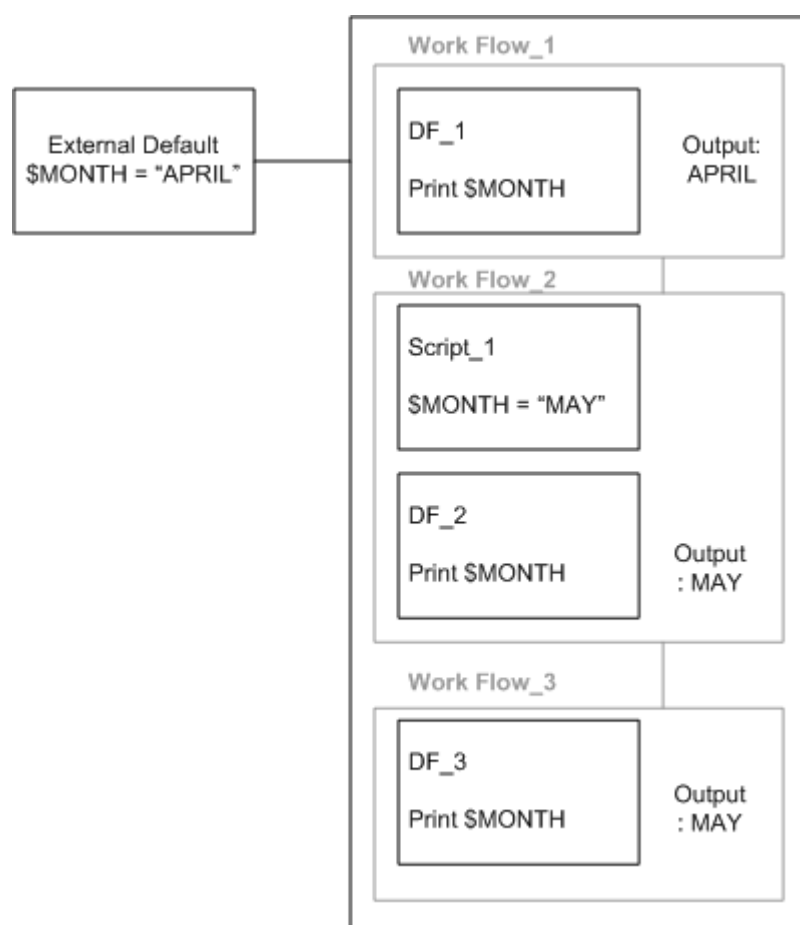
The value is determined at the time the software attempts to apply the external value.

A global variable value defined inside a job always overrides any external values. However, the override does not occur until the software attempts to apply the external values to the job being processed with the internal value. Up until that point, the software processes execution, schedule, or job property values as default values.

Example

Suppose you have a job called JOB_Test2 that has three work flows, each containing a data flow. The second data flow is inside a work flow that is preceded by a script in which \$MONTH is defined as 'MAY'. The first and third data flows have the same global variable with no value defined. The execution property \$MONTH = 'APRIL' is the global variable value.

In this scenario, 'APRIL' becomes the default value for the job. 'APRIL' remains the value for the global variable until it encounters the other value for the same variable in the second work flow. Since the value in the script is inside the job, 'MAY' overrides 'APRIL' for the variable \$MONTH. The software continues the processing the job with this new value.



22.8 Local and global variable rules

There are rules to follow when you set values for local and global variables.

The following table describes specific rules to follow when you set local and global variables.

Rule for	Description
Naming	- Local and global variables must have unique names within their job context. - Any name modification to a global variable can only be performed at the job level.
Replicating job and work flows	- When you replicate all objects, the local and global variables defined in that job context are also replicated. - When you replicate a data flow or work flow, all parameters and local and global variables are also replicated. However, you must validate these local and global variables within the job context in which they were created. i Note If you attempt to validate a data flow or work flow containing global variables without a job, Data Services reports an error.
Importing and exporting	- When you export a job object, you also export all local and global variables defined for that job. - When you export a lower-level object (such as a data flow) without the parent job, the global variable is not exported. Only the call to that global variable is exported. i Note If you use this object in another job without defining the global variable in the new job, a validation error will occur.

22.9 Environment variables

You can use system-environment variables inside jobs, work flows, or data flows. The `get_env`, `set_env`, and `is_set_env` functions provide access to underlying operating system variables that behave as the operating system allows.

You can temporarily set the value of an environment variable inside a job, work flow or data flow. Once set, the value is visible to all objects in that job.

Use the `get_env`, `set_env`, and `is_set_env` functions to set, retrieve, and test the values of environment variables.

Related Information

[Reference Guide: Functions and Procedures \[page 773\]](#)

22.10 Set file names at run-time using variables

You can set file names at runtime by specifying a variable as the file name.

Variables can be used as file names for these sources and targets:

- Flat files
- XML files and messages
- IDoc files and messages (in an SAP environment)

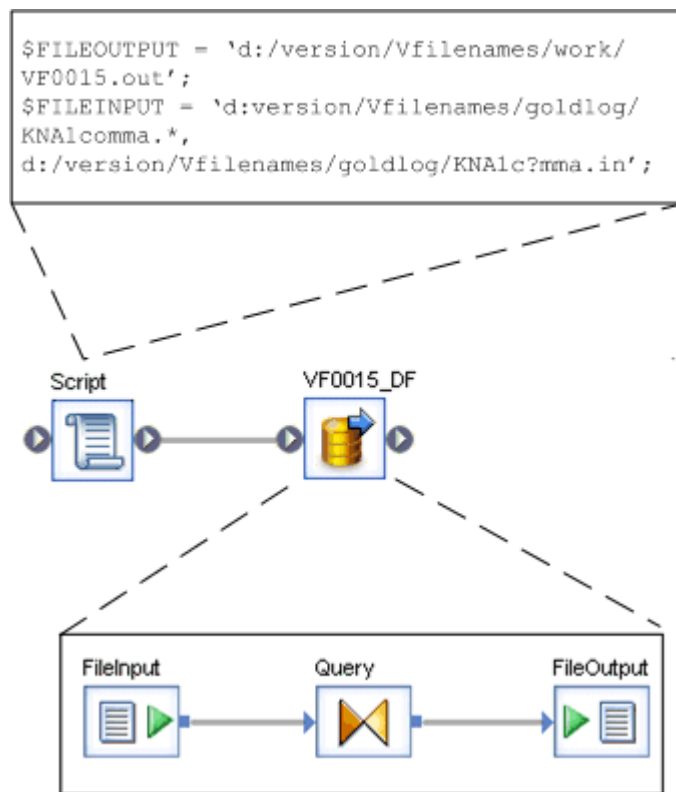
Variables can be also be used as file names for the `lookup_ext` function (for a flat file used as a lookup table parameter).

22.10.1 Using a variable in a flat file name

Set a flat file name using a global variable at runtime.

1. Create a local or global variable using **Tools > Variables** in the Variables and Parameters window. See [Defining a local variable \[page 710\]](#) or [Defining a global variable \[page 711\]](#) for details.
2. Create a script to set the value of a local or global variable following the steps in [Creating a script \[page 646\]](#). Or, call a system environment variable.
3. Declare the variable in the file format editor or in the Function editor as a `lookup_ext` parameter.
 - When you set a variable value for a flat file, specify both the file name and the directory name. Enter the variable in the *File(s)* property under *Data File(s)* in the File Format Editor. You cannot enter a variable in the *Root directory* property.
 - For lookups, substitute the path and file name in the *Lookup table* box in the `lookup_ext` function editor with the variable name.

The following figure shows how you can set values for variables in flat file sources and targets in a script.



When you use variables as sources and targets, you can also use multiple file names and wild cards. Neither is supported when using variables in the `lookup_ext` function.

The figure above provides an example of how to use multiple variable names and wild cards. Notice that the `$FILEINPUT` variable includes two file names (separated by a comma). The two names (`KNA1comma.*` and `KNA1c?mma.in`) also make use of the wild cards (`*` and `?`) supported by the software.

Related Information

[Reference Guide: Functions and Procedures, Descriptions of built-in functions, lookup_ext](#)

[Reference Guide: Scripting Language \[page 803\]](#)

22.11 Substitution parameters

22.11.1 Overview of substitution parameters

Use the same substitution parameter in multiple objects to efficiently set the same value in one location instead of editing each object separately.

When you use substitution parameters, you only need to edit the value once in the [Variables and Parameters](#) dialog to change the value in all objects that use the parameter. For example, if you create a job that references a unique directory on your local computer and you export that job to another computer, the job will look for the unique directory in the new environment. If that directory doesn't exist, the job won't run. When you use substitution parameters, the software automatically updates that value in all objects that use the substitution parameter.

For added convenience, you can configure a group of substitution parameters for a particular run-time environment by associating their constant values under a substitution parameter configuration. Create multiple configurations to group substitution parameters based on a run-time environment, for example. Then you can control many different values by changing configurations.

Use substitution parameters in expressions, SQL statements, option fields, and constant strings. Substitution parameters can appear as a value in many options and expression editors so that that you can choose a specific substitution variable from a list of values.

Example

Use a substitution parameter in a script such as:

```
Print('Data read in : [$$FilePath]'); or Print('[$$FilePath]');
```

Use a substitution parameter in a file format as the file name: [\$\$FilePath]/file.txt

Related Information

[Creating or renaming a configuration for a substitution parameter \[page 724\]](#)

[Data Quality default substitution parameters \[page 721\]](#)

22.11.1.1 Substitution parameters versus global variables

Substitution parameters differ from global variables in how they affect your jobs.

Substitution parameters apply at the repository level. Global variables apply only to the job in which they are defined. For example, you use a global variable for a parameter value that isn't calculated until the job is run,

whereas you use a substitution parameter for constants that do not change during execution. A substitution parameter defined in a given local repository is available to all the jobs in that repository. Therefore, using a substitution parameter means you do not need to define a global variable in each job to parameterize a constant value.

The following table describes the main differences between global variables and substitution parameters.

Global variables	Substitution parameters
Defined at the job level	Defined at the repository level
Cannot be shared across jobs	Available to all jobs in a repository
Data-type specific	No data type (all strings)
Value can change during job execution	Fixed value set prior to execution of job (constants)

You can use substitution parameters in all places where global variables are supported, for example, use substitution parameters in the following areas, which also support global variables:

- Query transform WHERE clauses
- Mappings
- SQL transform SQL statement identifiers
- Flat-file options
- User-defined transforms
- Address cleanse transform options
- Matching thresholds

22.11.1.2 Data Quality default substitution parameters

The software installs some default substitution parameters that are used by some Data Quality transforms.

For example, the following table lists default substitution parameters for the USA Regulatory Address Cleanse transform.

Substitution parameter	Description
\$\$RefFilesAddressCleanse	Defines the location of the address cleanse directories.
\$\$ReportsAddressCleanse	Enables data collection for creating reports with address cleanse statistics. Options are Yes or No.

Related Information

[Reference Guide, Transforms, USA Regulatory Address Cleanse, Reference files](#)

22.11.2 Tasks in the Substitution Parameter Editor

You can use the *Substitution Parameter Editor* to perform many tasks related to creating and editing substitution parameters and configurations.

Open the *Substitution Parameter Editor* from the Designer by selecting ► **Tools** ► *Substitution Parameter Configurations* .

What you can do in the Substitution Parameter Editor

Task	How to
Create new parameter	Right-click an existing substitution parameter that is next to where you want to place the new parameter and select <i>Insert</i> .
Copy an existing parameter	Right-click an existing substitution parameter and select <i>Copy</i> . i Note See Paste a copied parameter.
Paste a copied parameter	Right-click an existing substitution parameter that is below where you want to place the copied parameter, right click and select <i>Paste</i> . The pasted parameter appears in the line above the parameter you selected.
Edit an existing parameter	Double-click the existing parameter. A pencil icon appears to the left of the parameter name. Type your changes in the text box. When you have finished click on another row to view your change.
Filter the list of parameters for easier search	Enter the name or partial name of the parameter in the <i>Filter</i> text box and the list of parameters automatically displays a subset of the substitution parameters that match your filter. i Note You do not have to include the dollar signs as part of the filter. For example, enter "repository", not "\$\$repository".
View where used	Using the button: Select an existing parameter and click the <i>View Where Used</i> button Using the context menu: Right-click the existing parameter and select <i>View Where Used</i> from the context menu. The <i>Search</i> dialog opens showing a list of all objects that use the parameter.

Task	How to
Reorder parameter list	Select a parameter and drag it to a new position in the list.
Sort configurations	<i>i</i> Note You must have more than one configuration created. Select the <i>Sort Configuration Names in Ascending Order</i> or <i>Sort Configuration Names in Descending Order</i> icon from the tool bar.
Set a default configuration group	Click the drop arrow next to the <i>Default configuration</i> option and select a configuration name.
Create a new configuration group	Click the <i>Create New Substitution Parameter Configuration</i> icon in the tool bar.
Add a configuration to a substitution parameter	Double-click the configuration text box in the applicable parameter row. The pencil icon appears to the left of the parameter name. Type the new configuration.
Rename an existing configuration	Select the column heading of the applicable configuration and click the <i>Rename Substitution Parameter Configuration</i> icon in the tool bar. Type the new name.
Create a duplicate configuration	Select the applicable configuration and click the <i>Create a duplicate substitution Parameter Configuration</i> icon in the tool bar.
Delete a configuration	Select a configuration (cannot be a default configuration) column heading and click the <i>Delete Substitution Parameter Configuration</i> icon in the tool bar.
Move the designated default configuration to the front (the first column next to the Substitution Parameter list).	Click the <i>Move Default Configuration To Front</i> icon.

Related Information

[Creating a new substitution parameter \[page 724\]](#)

22.11.2.1 Naming substitution parameters

When you name and define substitution parameters, use the following rules:

- The name prefix is two dollar signs \$\$ (global variables are prefixed with one dollar sign). When adding new substitution parameters in the Substitution Parameter Editor, the editor automatically adds the prefix.

- When typing names in the Substitution Parameter Editor, do not use punctuation (including quotes or brackets) except underscores. The following characters are not allowed:

```
, : / ' \ " = < > + | - * % ; \t [ ] ( ) \r \n $ ] +
```

- You can type names directly into fields, column mappings, transform options, and so on. However, you must enclose them in square brackets, for example [\$\$SamplesInstall].
- Names can include any alpha or numeric character or underscores but cannot contain spaces.
- Names are not case sensitive.
- The maximum length for most repository types is 256 (MySQL is 64 and MS SQL server is 128).
- Names must be unique within the repository.

22.11.2.2 Creating a new substitution parameter

How to add a substitution parameter to a configuration.

You create substitution parameters to save time by entering the value once and saving it to a substitution parameter configuration. You set a constant value for the substitution parameter so that when you use the parameter in several objects, you only have to enter the constant value once, when you create it.

1. In the Designer, select  [Tools](#)  [Substitution Parameter Configurations](#) .

The [Substitution Parameter Editor](#) opens. The first column, named Substitution Parameter, lists the substitution parameters available in the repository. The second column contains the constant value for the parameter. By default the column heading is Configuration1.

2. Double-click a blank row. A pencil icon appears to the left of the row. Type a name. (You may have to scroll to the bottom of the Substitution Parameters column for a blank row.)

The software automatically adds a double dollar-sign prefix (\$\$) to the name when you navigate away from the cell.

3. Double-click in the next cell under the applicable configuration column and type the constant value that the parameter represents in that configuration. The software applies the constant value to the parameter when you run the job.

Related Information

[Creating or renaming a configuration for a substitution parameter \[page 724\]](#)

[Exporting and importing substitution parameters \[page 729\]](#)

22.11.2.3 Creating or renaming a configuration for a substitution parameter

How to create a new configuration group or rename an existing configuration group.

You may want to create a second configuration group so that you can assign more than one constant value to a substitution parameter. For example, if your directory files for testing a job are located in a different location

than the directories you use in production, you could assign two different constant values for the parameter. The software automatically names the first configuration group Configuration1, and subsequent configuration groups Configuration_1, Configuration_n+1, and so on.

1. To change the name of a configuration group, select the applicable configuration column heading to highlight it and click the *Rename Substitution Parameter Configuration* icon in the tool bar.
2. Type the new name in the highlighted column heading.

The new name appears in the column heading.

3. To add an additional configuration, click the *Create New Substitution Parameter Configuration* icon in the toolbar.

A new configuration column appears with the automatically-assigned name highlighted (for example, Configuration_1).

4. Type a unique name in the highlighted column heading.

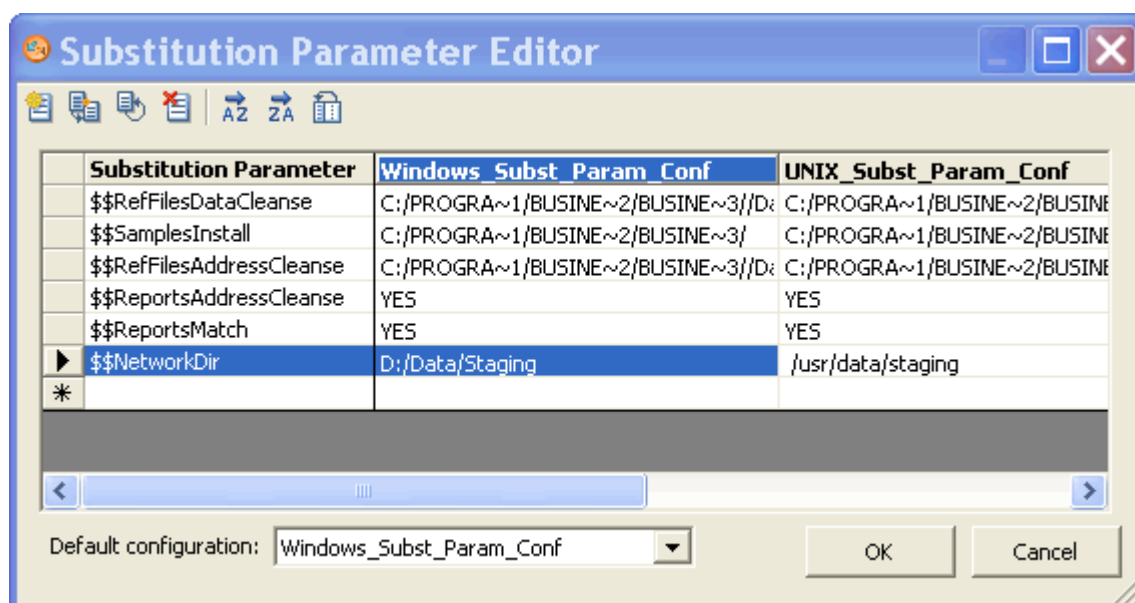
The software displays two configuration columns.

To change the default configuration that will apply when you run jobs, select it from the drop-down list box at the bottom of the window.

Example

The following screen capture shows two substitution parameter configuration groups. The substitution parameter \$\$NetworkDir has the value D:/Data/Staging in the configuration named Windows_Subst_Param_Conf and the value /usr/data/staging in the UNIX_Subst_Param_Conf configuration.

Notice that each configuration can contain multiple substitution parameters.



Related Information

[Substitution parameter configurations and system configurations \[page 726\]](#)

22.11.3 Substitution parameter configurations and system configurations

A system configuration groups together a set of datastore configurations and a substitution parameter configuration.

A substitution parameter configuration can be associated with one or more system configurations. For example, you might create one system configuration for your local system and a different system configuration for your remote system. Depending on your environment, both system configurations might point to the same substitution parameter configuration or each system configuration might require a different substitution parameter configuration.

At job execution time, you can set the system configuration and the job will execute with the values for the associated substitution parameter configuration.

Related Information

[Datastores \[page 92\]](#)

[Overview of substitution parameters \[page 720\]](#)

22.11.4 Associating substitution parameter configurations with system configurations

How to associate a substitution parameter configuration with a system configuration.

To associate a substitution parameter configuration with a system configuration:

1. In the Designer, open the *System Configuration Editor* by selecting **Tools** > *System Configurations* .

The editor contains at least two columns: The first column contains existing datastores, and the column is headed with the name Substitution Parameter. The second and subsequent columns are headed with the name of an existing system configuration. For example, if you have one system configuration, there is one column headed with the name of your system configuration.

2. In the row for the applicable datastore object, select the down arrow in the cell that is under the applicable system configuration column.

A list of existing substitution parameter configurations appears.

3. Select a substitution parameter configuration from the drop list.
4. If you want to create a new system configuration, click the *Create New Configuration* icon in the tool bar. A new column appears. Type the name of the new system configuration in the column heading.

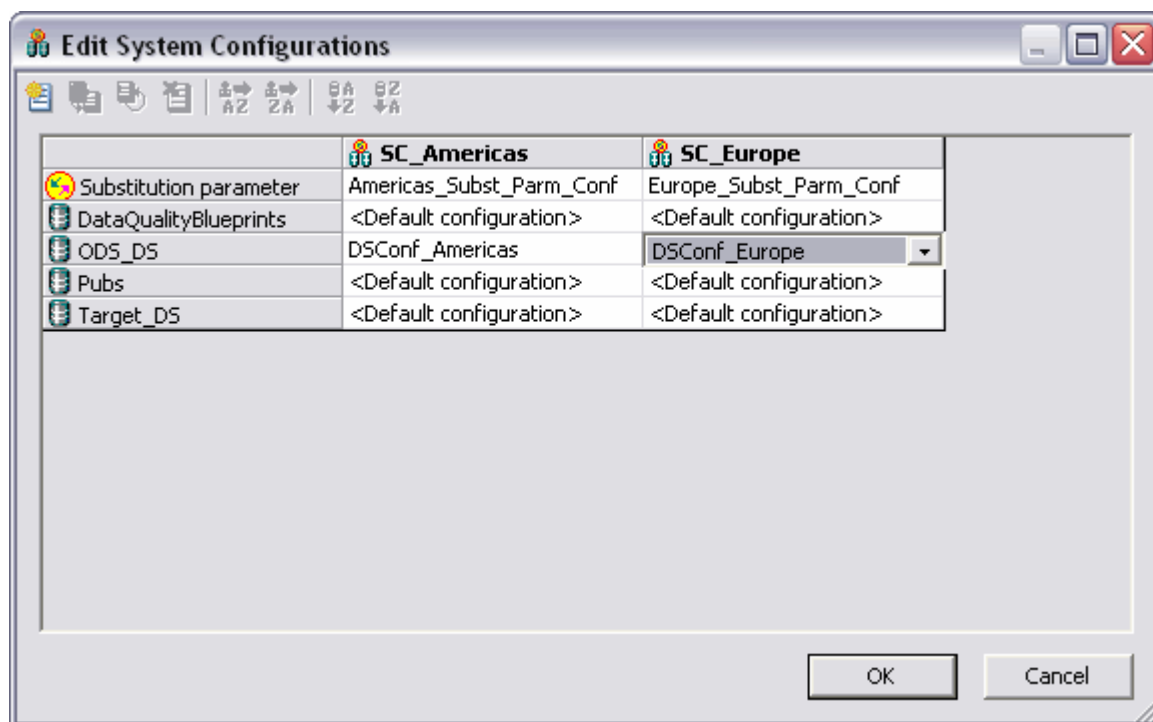
5. When you are finished, click [OK](#).

Example

The following example shows two system configurations, **SC_Americas** and **SC_Europe**. There are two substitution parameter configurations, one for each region: **Americas_Subst_Parm_Conf** and **Europe_Subst_Parm_Conf**. Each substitution parameter configuration defines where the data source files are located for that region. For example (not shown in the screen capture below):

- The data source files for the **Americas_Subst_Parm_Conf** are in D:/Data/Americas.
- The data source files for the **Europe_Subst_Parm_Conf** are in and D:/Data/Europe.

Select the appropriate substitution parameter configuration for each applicable datastore and system configuration. For example, the ODS_DS datastore (row under the first column, Substitution parameter) is associated with the DSConf_Americas substitution parameter configuration and the SC_Americas system configuration.



Related Information

[Define a system configuration \[page 216\]](#)

22.11.5 Overriding a substitution parameter in the Administrator

In the Administrator, you can override the substitution parameters, or select a system configuration to specify a substitution parameter configuration, on four pages:

- Execute Batch Job
- Schedule Batch Job
- Export Execution Command
- Real-Time Service Configuration

For example, the Execute Batch Job page displays the name of the selected system configuration, the substitution parameter configuration, and the name of each substitution parameter and its value.

To override a substitution parameter:

1. Select the appropriate system configuration.
2. Under *Substitution Parameters*, click *Add Overridden Parameter*, which displays the available substitution parameters.
3. From the drop-down list, select the substitution parameter to override.
4. In the second column, type the override value. Enter the value as a string without quotes (in contrast with Global Variables).
5. Execute the job.

22.11.6 Executing a job with substitution parameters

To see the details of how substitution parameters are being used in the job during execution in the Designer trace log:

1. Right-click the job name and click *Properties*.
2. Click the *Trace* tab.
3. For the *Trace Assemblers* option, set the value to *Yes*.
4. Click *OK*.

When you execute a job from the Designer, the Execution Properties window displays. You have the following options:

- On the *Execution Options* tab from the *System configuration* drop-down menu, optionally select the system configuration with which you want to run the job. If you do not select a system configuration, the software applies the default substitution parameter configuration as defined in the Substitution Parameter Editor. You can click *Browse* to view the *Select System Configuration* window in order to see the substitution parameter configuration associated with each system configuration. The *Select System Configuration* is read-only. If you want to change a system configuration, click **Tools** > *System Configurations* .
- You can override the value of specific substitution parameters at run time. Click the *Substitution Parameter* tab, select a substitution parameter from the Name column, and enter a value by double-clicking in the Value cell.

To override substitution parameter values when you start a job via a Web service, see the *Integrator Guide*.

Related Information

[Associating substitution parameter configurations with system configurations \[page 726\]](#)

[Overriding a substitution parameter in the Administrator \[page 728\]](#)

22.11.7 Exporting and importing substitution parameters

Substitution parameters are stored in a local repository along with their configured values. The software does not include substitution parameters as part of a regular export. You can, however, export substitution parameters and configurations to other repositories by exporting them to a file and then importing the file to another repository.

22.11.7.1 Exporting substitution parameters

1. Right-click in the local object library and select ► *Repository* ► *Export Substitution Parameter Configurations* .
2. Select the check box in the *Export* column for the substitution parameter configurations to export.
3. *Save* the file.
The software saves it as a text file with an .atl extension.

22.11.7.2 Importing substitution parameters

The substitution parameters must have first been exported to an ATL file.

Be aware of the following behaviors when importing substitution parameters:

- The software adds any new substitution parameters and configurations to the destination local repository.
 - If the repository has a substitution parameter with the same name as in the exported file, importing will overwrite the parameter's value. Similarly, if the repository has a substitution parameter configuration with the same name as the exported configuration, importing will overwrite all the parameter values for that configuration.
1. In the Designer, right-click in the object library and select ► *Repository* ► *Import from file* .
 2. Browse to the file to import.
 3. Click *OK*.

Related Information

[Exporting substitution parameters \[page 729\]](#)

23 Real-time Jobs

The software supports real-time data transformation.

Real-time means that the software can receive requests from ERP systems and Web applications and send replies immediately after getting the requested data from a data cache or a second application. You define operations for processing on-demand messages by building real-time jobs in the Designer.

23.1 Request-response message processing

Messages passed through a real-time system include the information required to perform a business transaction.

The content of the message can vary:

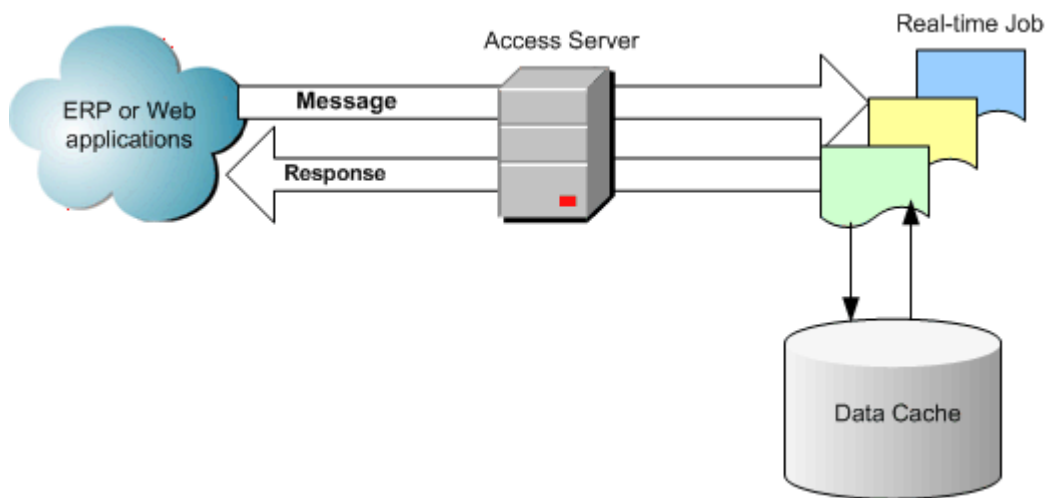
- It could be a sales order or an invoice processed by an ERP system destined for a data cache.
- It could be an order status request produced by a Web application that requires an answer from a data cache or back-office system.

The Access Server constantly listens for incoming messages. When a message is received, the Access Server routes the message to a waiting process that performs a predefined set of operations for the message type. The Access Server then receives a response for the message and replies to the originating application.

Two components support request-response message processing:

Component	Description
Access Server	Listens for messages and routes each message based on message type.
Real-time job	Performs a predefined set of operations for that message type and creates a response.

Processing might require that additional data be added to the message from a data cache or that the message data be loaded to a data cache. The Access Server returns the response to the originating application.



23.2 What is a real-time job?

The Designer allows you to define the processing of real-time messages using a real-time job.

You create a different real-time job for each type of message your system can produce.

23.2.1 Content of real-time job

A real-time processing loop can contain many objects in a single or multiple data flows.

A real-time processing loop can contain a single data flow, multiple data flows, or multiple work flows, scripts, conditionals, while loops, and so on.

- A single data flow can contain the following objects:
 - A single real-time source: XML message (required)
 - Sources: Files, XML files, and tables, including SAP tables
 - A single real-time target: XML message (required)
 - Targets: Files, XML files, tables, and template tables
 - Transforms, including queries
- Multiple data flows can contain the following objects:
 - A single real-time source in the first data flow: XML message (required)
 - Sources: Files, XML files, and tables, including SAP tables
 - A single real-time target in the last data flow: XML message (required)
 - Targets: Files, XML files, tables, and template tables
 - Transforms, including queries
 - Memory tables: Used as staging tables to move data sets to the next data flow in the job.
- Multiple work flows, scripts, conditionals, while loops, and so on:

You can also build real-time jobs using IDocs. See the *Supplement for SAP* for more information.

i Note

Real-time jobs can not use the following features: Data_Transfer transform, run as a separate process, or run at a job distribution level lower than a job level.

23.2.2 Real-time versus batch

Real-time jobs extract data from the body of the message received and from any secondary sources used in the job.

Like a batch job, a real-time job extracts, transforms, and loads data. Each real-time job can extract data from a single message type. It can also extract data from other sources such as tables or files.

The same powerful transformations you can define in batch jobs are available in real-time jobs. However, you might use transforms differently in real-time jobs. For example, you might use branches and logic controls more often than you would in batch jobs. If a customer wants to know when they can pick up their order at your distribution center, you might want to create a CheckOrderStatus job using a look-up function to count order items and then a case transform to provide status in the form of strings: "No items are ready for pickup" or "X items in your order are ready for pickup" or "Your order is ready for pickup".

Also in real-time jobs, the software writes data to message targets and secondary targets in parallel. This ensures that each message receives a reply as soon as possible.

Unlike batch jobs, real-time jobs do not execute in response to a schedule or internal trigger; instead, real-time jobs execute as real-time services started through the Administrator. Real-time services then wait for messages from the Access Server. When the Access Server receives a message, it passes the message to a running real-time service designed to process this message type. The real-time service processes the message and returns a response. The real-time service continues to listen and process messages on demand until it receives an instruction to shut down.

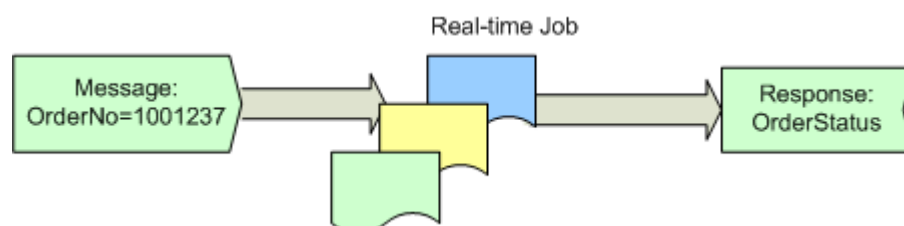
23.2.3 Messages

Typical messages include information required to implement a particular business operation and to produce an appropriate response.

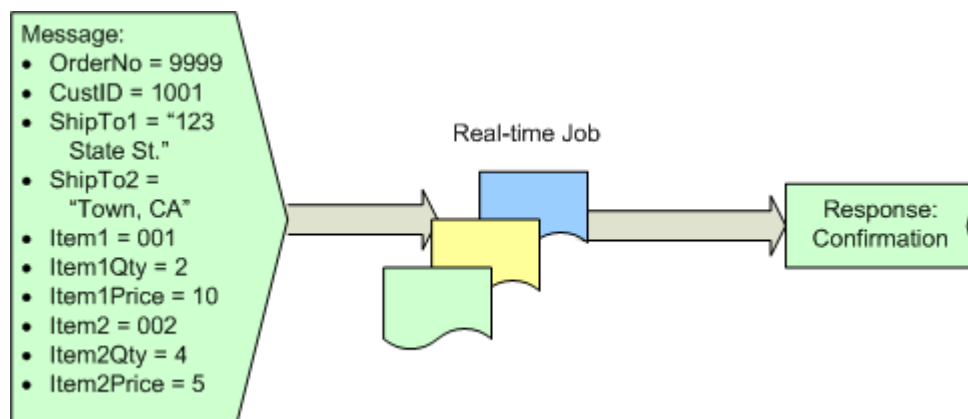
How you design a real-time job depends on what message you want it to process.

For example, suppose a message includes information required to determine order status for a particular order. The message contents might be as simple as the sales order number. The corresponding real-time job might use the input to query the right sources and return the appropriate product information.

In this case, the message contains data that can be represented as a single column in a single-row table.



In a second case, a message could be a sales order to be entered into an ERP system. The message might include the order number, customer information, and the line-item details for the order. The message processing could return confirmation that the order was submitted successfully.



In this case, the message contains data that cannot be represented in a single table; the order header information can be represented by a table and the line items for the order can be represented by a second table. The software represents the header and line item data in the message in a nested relationship.

Top-level table

OrderNo	CustID	ShipTo1	ShipTo2	LineItems
9999	1001	123 State St	Town, CA	

Nested table

Item	ItemQty	ItemPrice
001	2	10
002	4	5

When processing the message, the real-time job processes all of the rows of the nested table for each row of the top-level table. In this sales order, both of the line items are processed for the single row of header information.

Real-time jobs can send only one row of data in a reply message (message target). However, you can structure message targets so that all data is contained in a single row by nesting tables within columns of a single, top-level table.

The software data flows support the nesting of tables within other tables.

Related Information

[Nested Data \[page 651\]](#)

23.2.4 Message processing

The purpose of real-time jobs is to process multiple messages instead of files or tables as for batch jobs.

For transforms that require all of the message data at one time, such as queries that include aggregation functions, the software caches data temporarily. The transform performs the specified operation, then clears caches in preparation for the next message.

To test a real-time job, test one message and create a target test file to receive the data. A real-time job clears data after processing each message if you design the job using system defaults. To avoid having the software clear data after processing each message, disable the following options in the target editor:

- *Delete data from table before loading* for a table target
- *Delete file* for a flat file target
- *Delete and re-create file* for an XML target

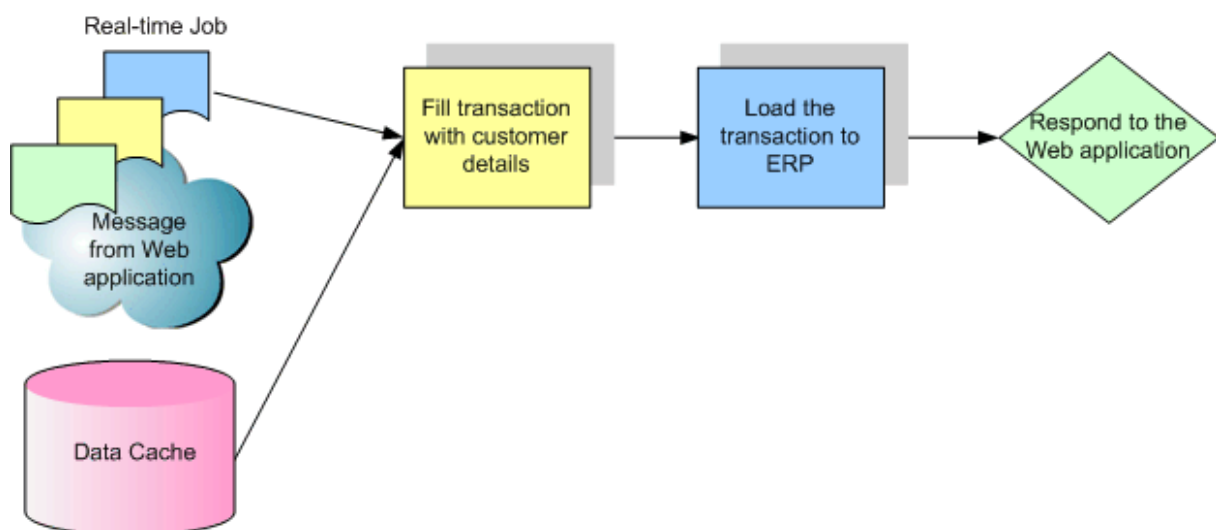
23.2.5 Real-time job examples

Example that provide a high-level description of how real-time jobs address typical real-time scenarios.

Later sections describe the actual objects that you would use to construct the logic in the Designer.

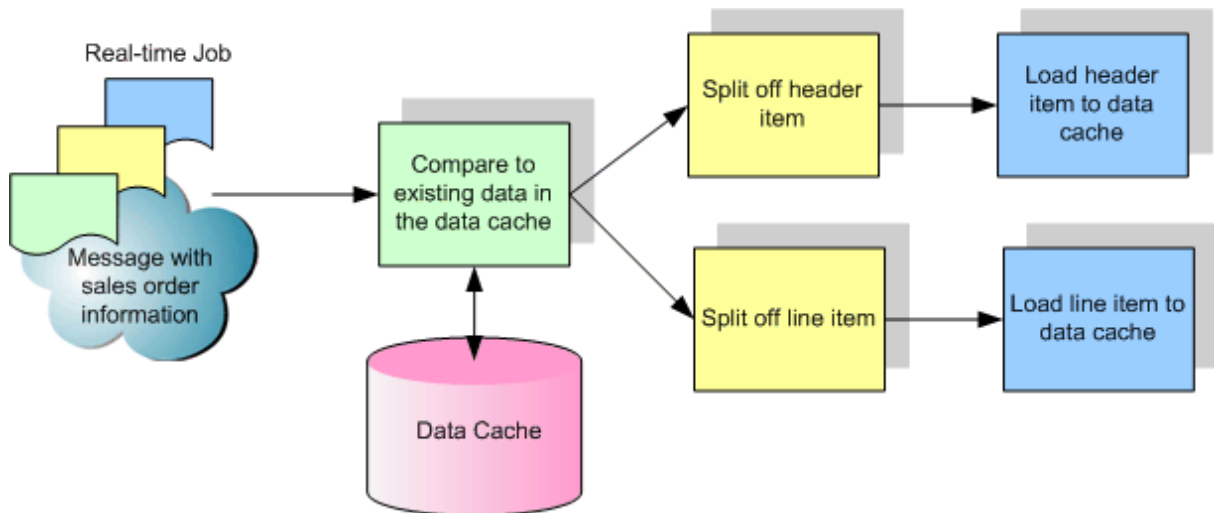
23.2.5.1 Loading transactions into a back-office application

A real-time job can receive a transaction from a Web application and load it to a back-office application (ERP, SCM, legacy). Using a query transform, you can include values from a data cache to supplement the transaction before applying it against the back-office application (such as an ERP system).



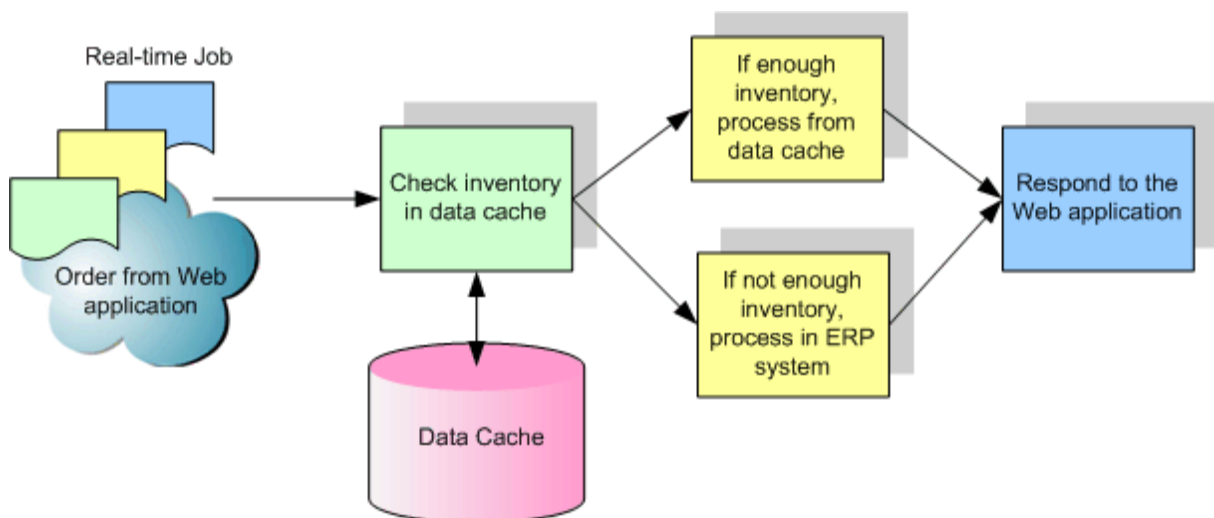
23.2.5.2 Collecting back-office data into a data cache

You can use messages to keep the data cache current. Real-time jobs can receive messages from a back-office application and load them into a data cache or data warehouse.



23.2.5.3 Retrieving values, data cache, back-office applications

You can create real-time jobs that use values from a data cache to determine whether or not to query the back-office application (such as an ERP system) directly.



23.3 Creating real-time jobs

You can create real-time jobs using the same objects as batch jobs (data flows, work flows, conditionals, scripts, while loops, etc.).

However, object usage must adhere to a valid real-time job model.

23.3.1 Arrange metadata

Insert any number of work flows and data flows to a real-time job by following data flow model processing loop.

If you delete an object from the object library that is also used in a real-time job, the software replaces calls to the object with an icon indicating that the calls are no longer valid.

You can insert any number of work flows and data flows into a real-time job as long as the data flow models for a real-time processing loop are followed:

- For a single data flow in a loop: Has both a message source and target.
- For multiple data flows in a loop: Has a message source in the first data flow and a message target in the last data flow.

At the job level, you cannot design work flows and data flows to run in parallel. However, you can design work flows and data flows to run in parallel inside a job level work flow.

The messages in a data flow are significant. The following table indicates how data flows can be used.

If a data flow has:	It can be used as the:
One XML message source	First data flow in a real-time processing loop.
One XML message target	Last data flow in a real-time processing loop.
One XML message source and one XML message target.	Only data flow in a real-time processing loop.
No message source or target	Not the first, last, or only data flow in a real-time processing loop. Data flow in a batch job.

Data flows that do not contain messages can be used in batch jobs or in real-time processing loops, between the first and last data flows.

23.3.2 Real-time job models

In contrast to batch jobs, which typically move large amounts of data at scheduled times, a real-time job, once started as a real-time service, listens for a request.

When a real-time job receives a request (typically to access small number of records), the software processes the request, returns a reply, and continues listening. This listen-process-listen logic forms a processing loop.

A real-time job is divided into three processing components: initialization, a real-time processing loop, and clean-up.

- The initialization component (optional) can be a script, work flow, data flow, or a combination of objects. It runs only when a real-time service starts.
- The real-time processing loop is a container for the job's single process logic. You can specify any number of work flows and data flows inside it.
- The clean-up component (optional) can be a script, work flow, data flow, or a combination of objects. It runs only when a real-time service is shut down.

In a real-time processing loop, a single message source must be included in the first step and a single message target must be included in the last step.

The following models support this rule:

- Single data flow model
- Multiple data flow model
- Request/Acknowledge data flow model

Related Information

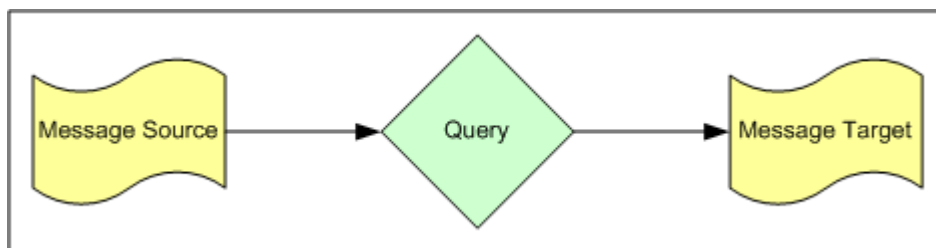
[Single data flow model \[page 737\]](#)

[Multiple data flow model \[page 737\]](#)

[Supplement for SAP: SAP ERP and R/3 and Real-Time Jobs, IDoc sources in real-time jobs](#)

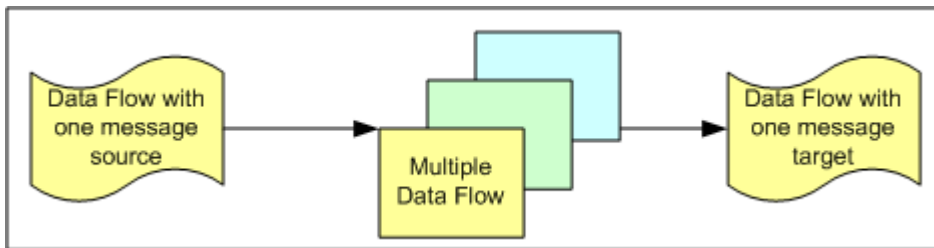
23.3.2.1 Single data flow model

With the single data flow model, you create a real-time job using a single data flow in its real-time processing loop. This single data flow must include a single message source and a single message target.



23.3.2.2 Multiple data flow model

The multiple data flow model allows you to create a real-time job using multiple data flows in its real-time processing loop.



By using multiple data flows, you can ensure that data in each message is completely processed in an initial data flow before processing for the next data flows starts. For example, if the data represents 40 items, all 40 must pass through the first data flow to a staging or memory table before passing to a second data flow. This allows you to control and collect all the data in a message at any point in a real-time job for design and troubleshooting purposes.

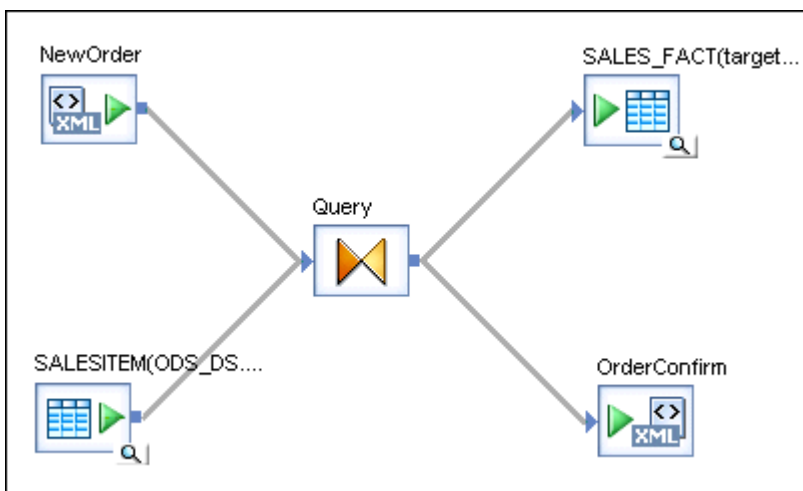
If you use multiple data flows in a real-time processing loop:

- The first object in the loop must be a data flow. This data flow must have one and only one message source.
- The last object in the loop must be a data flow. This data flow must have a message target.
- Additional data flows cannot have message sources or targets.
- You can add any number of additional data flows to the loop, and you can add them inside any number of work flows.
- All data flows can use input and/or output memory tables to pass data sets on to the next data flow. Memory tables store data in memory while a loop runs. They improve the performance of real-time jobs with multiple data flows.

23.3.3 Using real-time job models

23.3.3.1 Single data flow model

When you use a single data flow within a real-time processing loop your data flow diagram might look like this:



Notice that the data flow has one message source and one message target.

23.3.3.2 Multiple data flow model

When you use multiple data flows within a real-time processing loop your data flow diagrams might look like those in the following example scenario in which Data Services writes data to several targets according to your multiple data flow design.

Example scenario requirements:

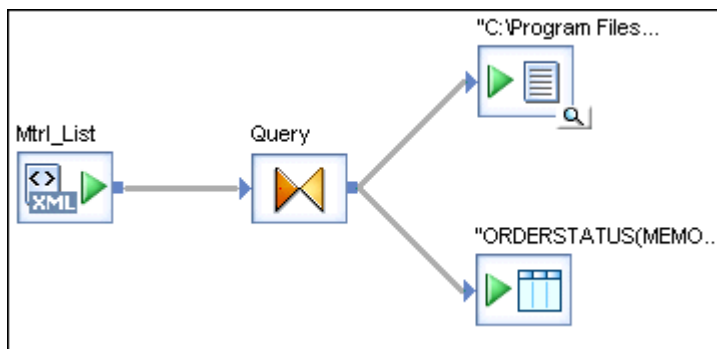
Your job must do the following tasks, completing each one before moving on to the next:

- Receive requests about the status of individual orders from a web portal and record each message to a backup flat file
- Perform a query join to find the status of the order and write to a customer database table.
- Reply to each message with the query join results

Solution:

First, create a real-time job and add a data flow, a work flow, and another data flow to the real-time processing loop. Second, add a data flow to the work flow. Next, set up the tasks in each data flow:

- The first data flow receives the XML message (using an XML message source) and records the message to the flat file (flat file format target). Meanwhile, this same data flow writes the data into a memory table (table target).

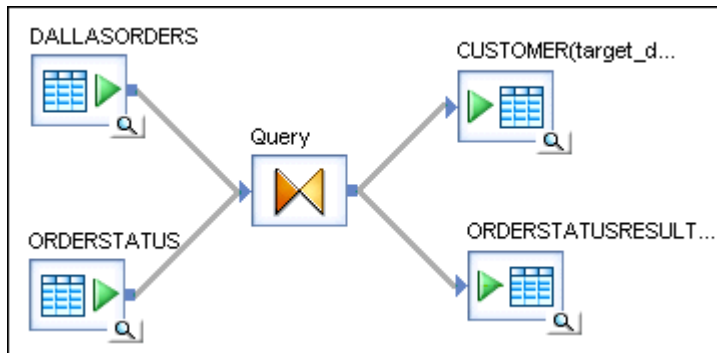


Note

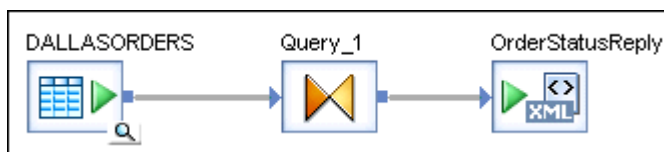
You might want to create a memory table to move data to sequential data flows.

- The second data flow reads the message data from the memory table (table source), performs a join with stored data (table source), and writes the results to a database table (table target) and a new memory table (table target).

Notice that this data flow has neither a message source nor a message target.



- The last data flow sends the reply. It reads the result of the join in the memory table (table source) and loads the reply (XML message target).



Related Information

[Designing real-time applications \[page 750\]](#)

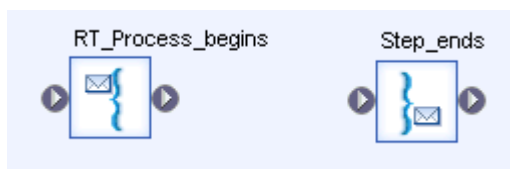
[Memory datastores \[page 136\]](#)

23.3.4 Creating a real-time job with a single data flow

This topic contains steps to create a real-time job using a single data flow.

1. In Designer, create a new project, or open an existing project.
2. Right-click an empty area of the Project Area and select *New Real-time job* from the dropdown menu.

A new job icon appears in the Project Area named *New_RTJob1*. The workspace displays the project structure, which consists of a begin and end marker.



These markers represent the beginning and end of a real-time processing loop.

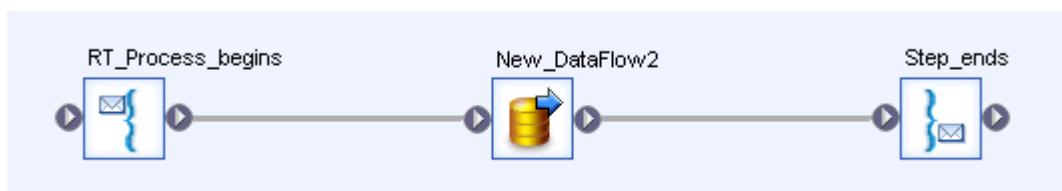
3. Rename *New_RTJob1*.

It is easier to find your real time objects by adding a prefix to the object name. For example, when naming this job, begin the name with the prefix `RT_Job_<projectname>`.

The software lists all saved real-time jobs under the Job tab in the Local Object Library alphabetically. Therefore all of your real-time jobs are grouped together. However, job names may also appear in text

editors that are used to create adapter or Web Services calls. In these cases, a prefix saved with the job name will help you identify it.

4. Add a data flow to the real-time job workspace using the tool pallet.
5. Connect the objects of the job in the workspace with the data flow as the middle object.



6. Build the data flow including a source and target.
7. Add, configure, and connect initialization object(s) and clean-up object(s) as needed.

Related Information

[Creating and defining data flows \[page 275\]](#)

[Real-time source and target objects \[page 741\]](#)

23.4 Real-time source and target objects

Real-time jobs must contain a real-time source and/or target object.

Those normally available are:

Object	Description	Used as a:	Software access
XML message	An XML message structured in a DTD or XML Schema format	Source or target	Directly or through adapters
Outbound message	A real-time message with an application-specific format (not readable by XML parser)	Target	Through an adapter

You can also use IDoc messages as real-time sources for SAP applications. For more information, see the *Supplement for SAP*.

Adding sources and targets to real-time jobs is similar to adding them to batch jobs, with the following additions:

For	Prerequisite	Object library location
XML messages	Import a DTD or XML Schema to define a format	Formats tab

For	Prerequisite	Object library location
Outbound message	Define an adapter datastore and import object metadata.	Datastores tab, under adapter datastore

Related Information

[Importing a DTD or XML Schema format \[page 671\]](#)

23.4.1 Viewing an XML message source or target schema

In the workspace of a real-time job, click the name of an XML message source or XML message target to open its editor.

If the XML message source or target contains nested data, the schema displays nested tables to represent the relationships among the data.

23.4.2 Load targets as a single transaction

For real-time jobs, load more than one table from a single datastore in a single transaction.

When you enable transactional loading for table targets, SAP Data Services sends `INSERT` statements for any of the tables included in the transaction to the database to process. You can also control which tables are loaded first by specifying the transaction order for the tables.

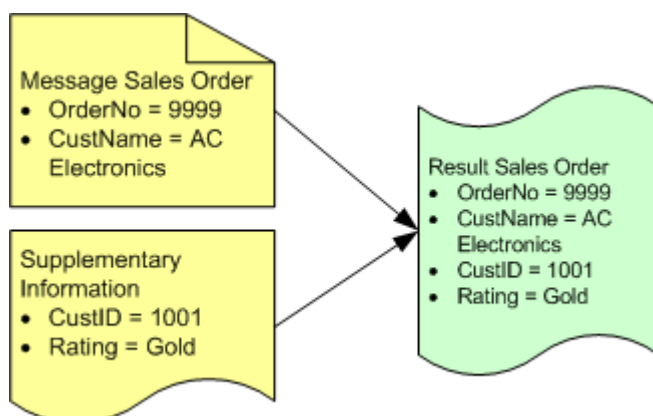
If the data flow includes a real-time target, it is always loaded in parallel with other targets to ensure that load time is as short as possible.

23.4.3 Secondary sources and targets

Real-time jobs can also have secondary sources or targets.

For example, suppose you are processing a message that contains a sales order from a Web application. The order contains the customer name, but when you apply the order against your ERP system, you need to supply more detailed customer information.

Inside a data flow of a real-time job, you can supplement the message with the customer information to produce the complete document to send to the ERP system. The supplementary information might come from the ERP system itself or from a data cache containing the same information.



Tables and files (including XML files) as sources can provide this supplementary information.

The software reads data from secondary sources according to the way you design the data flow. The software loads data to secondary targets in parallel with a target message.

Add secondary sources and targets to data flows in real-time jobs as you would to data flows in batch jobs.

Related Information

[Source and target objects \[page 277\]](#)

23.4.4 Transactional loading of tables

Target tables in real-time jobs support transactional loading, in which the data resulting from the processing of a single data flow can be loaded into multiple tables as a single transaction.

No part of the transaction applies if any part fails.

i Note

Target tables in batch jobs also support transactional loading. However, use caution when you consider enabling this option for a batch job because it requires the use of memory, which can reduce performance when moving large amounts of data.

You can specify the order in which tables in the transaction are included using the target table editor. This feature supports a scenario in which you have a set of tables with foreign keys that depend on one with primary keys.

You can use transactional loading only when all the targets in a data flow are in the same datastore. If the data flow loads tables in more than one datastore, targets in each datastore load independently. While multiple targets in one datastore may be included in one transaction, the targets in another datastores must be included in another transaction.

You can specify the same transaction order or distinct transaction orders for all targets to be included in the same transaction. If you specify the same transaction order for all targets in the same datastore, the tables are

still included in the same transaction but are loaded together. Loading is committed after all tables in the transaction finish loading.

If you specify distinct transaction orders for all targets in the same datastore, the transaction orders indicate the loading orders of the tables. The table with the smallest transaction order is loaded first, and so on, until the table with the largest transaction order is loaded last. No two tables are loaded at the same time. Loading is committed when the last table finishes loading.

23.4.5 Design tips for data flows in real-time jobs

When designing data flows in real-time jobs, there are rules that need to be followed.

- If you include a table in a join with a real-time source, the software includes the data set from the real-time source as the outer loop of the join. If more than one supplementary source is included in the join, you can control which table is included in the next outer-most loop of the join using the join ranks for the tables.
- In real-time jobs, do not cache data from secondary sources unless the data is static. The data will be read when the real-time job starts and will not be updated while the job is running.
- If no rows are passed to the XML target, the real-time job returns an empty response to the Access Server. For example, if a request comes in for a product number that does not exist, your job might be designed in such a way that no data passes to the reply message. You might want to provide appropriate instructions to your user (exception handling in your job) to account for this type of scenario.
- If more than one row passes to the XML target, the target reads the first row and discards the other rows. To avoid this issue, use your knowledge of the software's Nested Relational Data Model (NRDM) and structure your message source and target formats so that one "row" equals one message. With NRDM, you can structure any amount of data into a single "row" because columns in tables can contain other tables.
- Recovery mechanisms are not supported in real-time jobs.

Related Information

[Reference Guide: Objects, Descriptions of objects, Real-time job Nested Data \[page 651\]](#)

23.5 Testing real-time jobs

23.5.1 Executing a real-time job in test mode

You can test real-time job designs without configuring the job as a service associated with an Access Server.

In test mode, you can execute a real-time job using a sample source message from a file to determine if the software produces the expected target message.

23.5.1.1 Specifying a sample XML message and target test file

1. In the XML message source and target editors, enter a file name in the *XML test file* box.

Enter the full path name for the source file that contains your sample data. Use paths for both test files relative to the computer that runs the Job Server for the current repository.

2. Execute the job.

Test mode is always enabled for real-time jobs. The software reads data from the source test file and loads it into the target test file.

23.5.2 Using View Data

Use the View Data feature to capture a sample of your output data to ensure that your design is working.

This allows you to see that your design returns the results you expect.

Related Information

[Design and Debug \[page 815\]](#)

23.5.3 Using an XML file target

Use an XML file target to capture the message produced by a data flow while allowing the message to be returned to the Access Server.

Just like an XML message, you define an XML file by importing a DTD or XML Schema for the file, then dragging the format into the data flow definition. Unlike XML messages, you can include XML files as sources or targets in batch and real-time jobs.

23.5.3.1 Using a file to capture output from steps in a real-time job

1. In the *Formats* tab of the object library, drag the DTD or XML Schema into a data flow of a real-time job.

A menu prompts you for the function of the file.

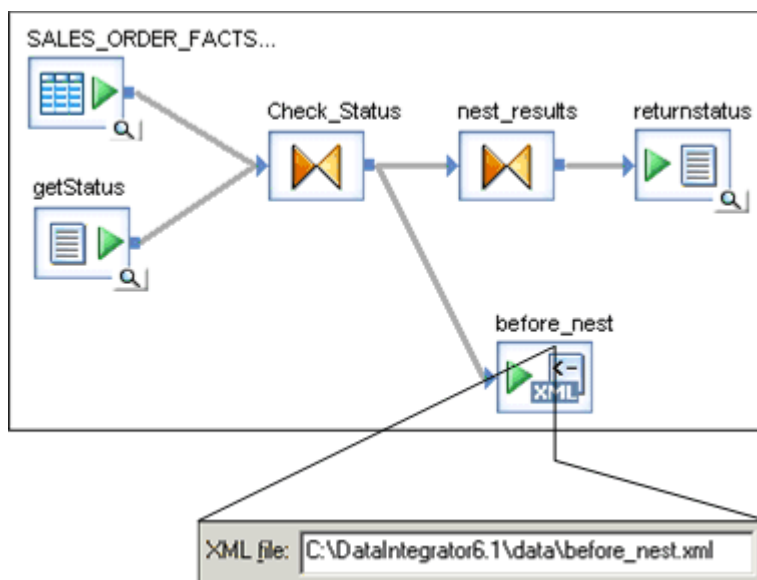
2. Choose *Make XML File Target*.

The XML file target appears in the workspace.

3. In the file editor, specify the location to which the software writes data.

Enter a file name relative to the computer running the Job Server.

4. Connect the output of the step in the data flow that you want to capture to the input of the file.



23.6 Building blocks for real-time jobs

23.6.1 Supplementing message data

You can define steps in the real-time job to supplement message information if the data included in messages from real-time sources does not map exactly to your requirements for processing or storing the information.

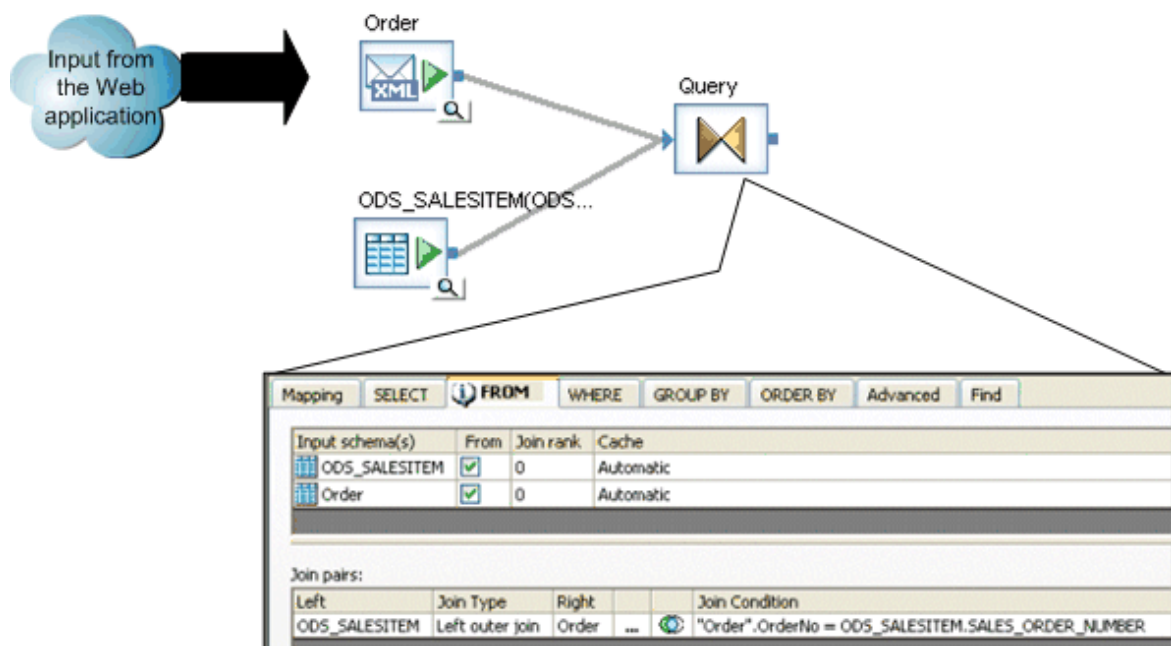
One technique for supplementing the data in a real-time source includes these steps:

1. Include a table or file as a source.

In addition to the real-time source, include the files or tables from which you require supplementary information.

2. Use a query to extract the necessary data from the table or file.
3. Use the data in the real-time source to find the necessary supplementary data.

You can include a join expression in the query to extract the specific values required from the supplementary source.



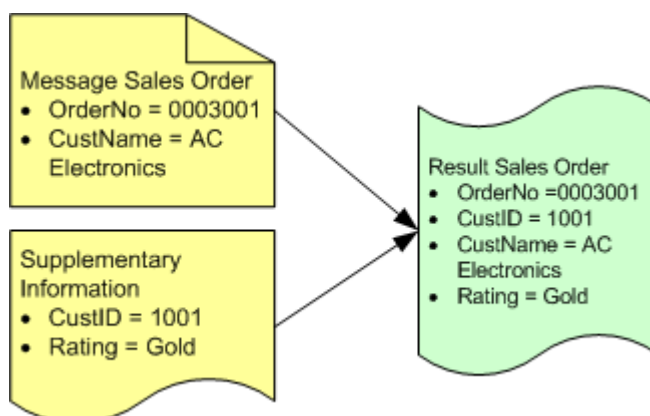
The Join Condition joins the two input schemas resulting in output for only the sales item document and line items included in the input from the application.

Be careful to use data in the join that is guaranteed to return a value. If no value returns from the join, the query produces no rows and the message returns to the Access Server empty. If you cannot guarantee that a value returns, consider these alternatives:

- [Lookup function call](#)—Returns a default value if no match is found.
- [Outer join](#)—Always returns a value, even if no match is found.

23.6.1.1 Supplementing message data example

In this example, a request message includes sales order information and its reply message returns order status. The business logic uses the customer number and priority rating to determine the level of status to return. The message includes only the customer name and the order number. A real-time job is then defined to retrieve the customer number and rating from other sources before determining the order status.



1. Include the real-time source in the real-time job.
2. Include the supplementary source in the real-time job.

This source could be a table or file. In this example, the supplementary information required doesn't change very often, so it is reasonable to extract the data from a data cache rather than going to an ERP system directly.

3. Join the sources.

In a query transform, construct a join on the customer name:

```
Message.CustName = Cust_Status.CustName
```

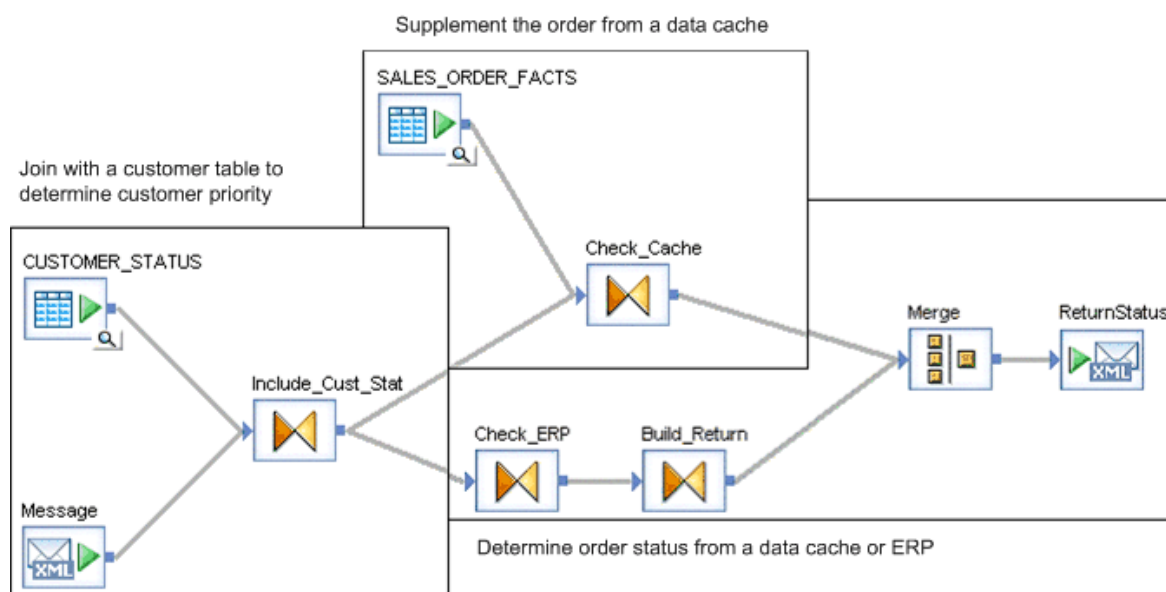
You can construct the output to include only the columns that the real-time job needs to determine order status.

4. Complete the real-time job to determine order status.

The example shown here determines order status in one of two methods based on the customer status value. Order status for the highest ranked customers is determined directly from the ERP. Order status for other customers is determined from a data cache of sales order information.

The logic can be arranged in a single or multiple data flows. The illustration below shows a single data flow model.

Both branches return order status for each line item in the order. The data flow merges the results and constructs the response. The next section describes how to design branch paths in a data flow.



23.6.2 Branching data flow based on a data cache value

One of the most powerful things you can do with a real-time job is to design logic that determines whether responses should be generated from a data cache or if they must be generated from data in a back-office application (ERP, SCM, CRM).

Here is one technique for constructing this logic:

1. Determine the rule for when to access the data cache and when to access the back-office application.
2. Compare data from the real-time source with the rule.
3. Define each path that could result from the outcome.

You might need to consider the case where the rule indicates back-office application access, but the system is not currently available.

4. Merge the results from each path into a single data set.
5. Route the single result to the real-time target.

You might need to consider error-checking and exception-handling to make sure that a value passes to the target. If the target receives an empty set, the real-time job returns an empty response (begin and end XML tags only) to the Access Server.

This example describes a section of a real-time job that processes a new sales order. The section is responsible for checking the inventory available of the ordered products—it answers the question, "is there enough inventory on hand to fill this order?"

The rule controlling access to the back-office application indicates that the inventory (Inv) must be more than a pre-determined value (IMargin) greater than the ordered quantity (Qty) to consider the data cached inventory value acceptable.

The software makes a comparison for each line item in the order they are mapped.

Incoming sales order

OrderNo	CustID	LineItem		
		Item	Material	Qty
9999	1001	001	7333	300
		002	2288	1400

Inventory data cache

Material	Inv	IMargin
7333	600	100
2288	1500	200

i Note

The quantity of items in the sales order is compared to inventory values in the data cache.

23.6.3 Calling application functions

A real-time job can use application functions to operate on data.

You can include tables as input or output parameters to the function.

Application functions require input values for some parameters and some can be left unspecified. You must determine the requirements of the function to prepare the appropriate inputs.

To make up the input, you can specify the top-level table, top-level columns, and any tables nested one-level down relative to the tables listed in the FROM clause of the context calling the function. If the application function includes a structure as an input parameter, you must specify the individual columns that make up the structure.

A data flow may contain several steps that call a function, retrieve results, then shape the results into the columns and tables required for a response.

23.7 Designing real-time applications

You can design real-time applications that meet your internal and external information and resource needs.

The software provides a reliable and low-impact connection between a Web application and back-office applications such as an enterprise resource planning (ERP) system. Because each implementation of an ERP system is different and because the software includes versatile decision support logic, you may decide to design a system.

23.7.1 Reducing queries requiring back-office application access

A collection of recommendations and considerations that can help reduce the time you spend experimenting in your development cycles.

The information you allow your customers to access through your Web application can impact the performance that your customers see on the Web. You can maximize performance through your Web application design decisions. In particular, you can structure your application to reduce the number of queries that require direct back-office (ERP, SCM, Legacy) application access.

For example, if your ERP system supports a complicated pricing structure that includes dependencies such as customer priority, product availability, or order quantity, you might not be able to depend on values from a data cache for pricing information. The alternative might be to request pricing information directly from the ERP system. ERP system access is likely to be much slower than direct database access, reducing the performance your customer experiences with your Web application.

To reduce the impact of queries requiring direct ERP system access, modify your Web application. Using the pricing example, design the application to avoid displaying price information along with standard product information and instead show pricing only after the customer has chosen a specific product and quantity. These techniques are evident in the way airline reservations systems provide pricing information—a quote for a

specific flight—contrasted with other retail Web sites that show pricing for every item displayed as part of product catalogs.

23.7.2 Messages from real-time jobs to adapter instances

If a real-time job will send a message to an adapter instance, you'll need to decide if you need to create a message function call or an outbound message.

- Message function calls allow the adapter instance to collect requests and send replies.
- Outbound message objects can only send outbound messages. They cannot be used to receive messages.

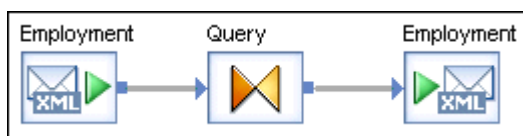
23.7.3 Real-time service invoked by an adapter instance

When an operation instance (in an adapter) gets a message from an information resource, it translates it to XML (if necessary), then sends the XML message to a real-time service.

In the real-time service, the message from the adapter is represented by a DTD or XML Schema object (stored in the [Formats](#) tab of the object library). The DTD or XML Schema represents the data schema for the information resource.

The real-time service processes the message from the information resource (relayed by the adapter) and returns a response.

In the example data flow below, the Query processes a message (here represented by "Employment") received from a source (an adapter instance), and returns the response to a target (again, an adapter instance).



i Note

This document uses terms consistent with Java programming. (Please see your adapter SDK documentation for more information about terms such as operation instance and information resource.)

23.7.4 Start and stop real-time services

Start and stop services based on whether you run in test or development mode.

For development and testing, you can manually execute a real-time job from the Designer. The Designer runs a real-time job in test mode.

When testing a real-time service or when running in production, the Access Server triggers the Job Server to process a request using the logic you built inside a real-time processing loop. The Access Server can also

trigger the Job Server to shut down real-time processing loops. In a production environment, you control the operation of real-time services using the Administrator.

24 Executing Jobs

This section contains an overview of the software job execution, steps to execute jobs, debug errors, and change job server options.

24.1 Overview of job execution

You can run jobs in three different ways. Depending on your needs, you can configure:

Job	Description
Immediate jobs	The software initiates both batch and real-time jobs and runs them immediately from within the Designer. For these jobs, both the Designer and designated Job Server (where the job executes, usually many times on the same machine) must be running. You will most likely run immediate jobs only during the development cycle.
Scheduled jobs	Batch jobs are scheduled. To schedule a job, use the Administrator or use a third-party scheduler. When jobs are scheduled by third-party software: • The job initiates outside of the software. • The job operates on a batch job (or shell script for UNIX) that has been exported from the software. When a job is invoked by a third-party scheduler: • The corresponding Job Server must be running. • The Designer does not need to be running.
Services	Real-time jobs are set up as services that continuously listen for requests from an Access Server and process requests on-demand as they are received. Use the Administrator to create a service from a real-time job.

24.2 Preparing for job execution

24.2.1 Validate jobs and job components

SAP Data Services provides tools to validate and fix errors in a job during design and before you execute the job.

Validate during job creation

As you add objects to a data flow within your job, validate the job after adding and configuring objects. Use one of the validate icons in the toolbar. The following table describes each option.

 <i>Validate All</i>	Validates syntax for object definitions for all objects that are in the current workspace, recursively. Menu equivalent: ► <i>Validate</i> ► <i>All Objects in View</i> ►
 <i>Validate Current View</i>	Validates the syntax of the object definition for the active workspace. For example, validates syntax in the Query Editor when you have the Query Editor open in the workspace. Menu equivalent: ► <i>Validate</i> ► <i>Current View</i> ►

Automatic validation before job execution

Set Designer to always validate a job before job execution by checking the option *Perform complete validation before job execution*. The default setting is not checked.

1. Select ► *Tools* ► *Options* ►.
2. Expand *Designer* and select *General*. The option is included in the list of options.

If you don't select this option, Data Services validates the job during job execution and stops processing when it encounters an error.

Validation errors

When Data Services discovers an error in an object definition, it opens the *Output* dialog box with error information.

If you can't see the complete error text in the *Output* dialog box, right-click the error text and select *View*.

To fix an error, double-click the error in the *Output* dialog box to open the object editor. For example, if the error exists in a target table, the Target Editor opens.

The following table contains descriptions for all the possible icons that appear in execution messages. The icons also appear during job execution as you view the job execution progress.

Icon	Description
 <i>Information</i>	Information about the job execution process. The information content doesn't prevent the job from running. No action required.
 <i>Warning</i>	Information about an issue that may cause unexpected results after execution. The situation that caused the warning isn't severe enough to stop job execution. However, if you fix the situation that caused the warning, you can prevent any unexpected results.
 Example For example, a warning may tell you that the data type of a source column doesn't match the data type of the target column.	
 <i>Error</i>	Information about an issue that prevents the job to execute. Fix the error before you execute the job.

24.2.2 Ensuring that the Job Server is running

Before you execute a job, either as an immediate or scheduled task, ensure that the Job Server is running.

With the SAP Data Services Designer open, look at the bottom right of the page. The status of the Job Server is indicated with icons.

Icon	Description
	Job Server is running
	Job Server is inactive

The name of the active Job Server and port number appears in the status bar in the lower left when the cursor is over the icon.

An additional icon appears indicating whether the profile server is running. You will know what the icon represents by viewing the message that appears when you hover your mouse over the icon.

24.2.3 Setting job execution options

Options for jobs include Debug and Trace. Although these are object options—they affect the function of the object—they are located in either the Property or the Execution window associated with the job.

Execution options for jobs can either be set for a single instance or as a default value.

- The right-click [Execute](#) menu sets the options for a single execution only and overrides the default settings.
- The right-click [Properties](#) menu sets the default settings.

24.2.3.1 Execution Options

Use execution options to help capture and diagnose errors using log, auditing, statistics collection, or recovery options.

Log information goes to one of four files in the `<DS_COMMON_DIR>\log\<Job_Server_Name>\<repository name>` directory:

- Error log file
- Monitor log file
- Trace log file
- Debug package file

When you execute a job, you can also select a system configuration and a Job Server or server group from the [Execution Options](#) tab of the [Execution Properties](#) window.

Select the [Execution Options](#) tab to set the following options.

Options	Description
Collect statistics for monitoring	Select to display cache statistics in the Performance Monitor, located in the Management Console Administrator. Default setting: Not selected. i Note Use this option only to look at the cache size. For more information, see the Using Caches section in the <i>Performance Optimization Guide</i>.
Collect statistics for optimization	Select to collect statistics that the Data Services optimizer uses to choose an optimal cache type: In-memory and pageable. Default setting: Not selected. For more information, see the <i>Performance Optimization Guide</i>.
Disable data validation statistics collection	Select when you do not want to collect data validation statistics for any validation transforms in the job. Default setting: Not selected. For more information about data validation statistics, see the <i>Management Console Guide</i>.

Options	Description
Distribution level	Select the level with which to distribute job processing to multiple job servers. Options include the following: • Job: The whole job executes on an available Job Server. • Data flow: Each data flow within the job executes on an available Job Server. • Secondary data flow: Each secondary data flow, either a separate transform or a function, within a data flow executes on an available Job Server. For more information, see “Using grid computing to distribute data flows execution” and “Distributing Data Flow Execution” in the <i>Performance Optimization Guide</i>.
Enable auditing	Select to collect audit statistics for this specific job execution. Default setting: Selected. For more information about auditing, see the <i>Designer Guide</i>.
Enable recovery	Applies to batch jobs only. Select to enable the automatic recovery feature. When enabled, the software saves the results from completed steps and allows you to resume failed jobs. Default setting: Not selected. You cannot enable the automatic recovery feature when executing a job in data scan mode. Also, this option is available only as a runtime property. It is not available as a default property. For more information, see the <i>Designer Guide</i>.
Export Data Quality reports	Select to generate and export all specified job reports to the location specified in the Report Server Configuration node in the Management Console Administrator. Default setting: The reports are exported to <code><DS_COMMON_DIR>\DataQuality\reports\<repository>\job<></code>.
Job Server or Server group	Select either a Job Server or a server group to execute the job. The list contains Job Servers and server groups linked to the job repository. Define the Job Server with a host name and port. Define a server group by the server group name. When you select a Job Server or server group, remember that many objects in the Designer have options set relative to the Job Server location. For example: • Directory and file names for source and target files • Bulk load directories For more information, see the <i>Management Console Guide</i>.
Monitor sample rate (# of seconds)	Enter the number of seconds elapsed before the software writes information to the monitor log file and updates the job events. The software writes information about the status of each source, target, or transform. Default setting: 5 seconds. For example, if you enter 30, the logs update every 30 seconds. When you set the value, evaluate performance improvements gained by making fewer calls to the operating system against your ability to find errors quickly. With a higher monitor sample rate, more data collects before calling the operating system to open the file: performance improves. However, with a higher monitor rate, more time passes before you are able to see any errors. Setting the rate to 0 disables the monitoring feature. i Note If you use a virus scanner on your files, exclude the log from the virus scan. Otherwise, the virus scan analyzes the log repeated during the job execution, which causes a performance degradation.

Options	Description
Print all trace messages	Select to print all trace messages to the trace log file for the current Job Server. Selecting this option overrides the trace properties set on the Trace tab.
Recover from last failed execution	Applies to batch jobs only. Select to resume a failed job. The software retrieves the results from any steps that were previously executed successfully and re-executes any other steps. This option is a runtime property and is only available if the job hasn't been executed or if you disabled recovery mode during a previous run. For more information about automatic recovery, see the <i>Designer Guide</i>.
System configuration	Select the system configuration to use when executing this job. A system configuration defines a set of datastore configurations, which define the datastore connections. If a system configuration is not specified, the software uses the default datastore configuration for each datastore. This option is a runtime property. This option is available only if there are system configurations defined in the repository. For more information about creating and managing multiple datastore configurations, see the <i>Designer Guide</i>.
Use collected statistics	Select to have the optimizer use the cache statistics collected on a previous execution of the job. Default setting: Selected. For more information, see the <i>Performance Optimization Guide</i>.
Single instance	Select to prevent multiple instances of a job running.

24.2.3.2 Setting execution options for every execution of the job

1. From the [Project](#) area, right-click the job name and choose [Properties](#).
2. Select options on the Properties window.

Related Information

[Adding, changing, and viewing object properties \[page 71\]](#)

[Trace properties \[page 764\]](#)

[Setting a global variable value as a job property \[page 713\]](#)

[Setting a global variable value as an execution property \[page 714\]](#)

[Execution Options \[page 756\]](#)

24.2.3.3 Limiting a job to a single instance

Ensure that the software executes only one instance of the job.

When *Single instance* is enabled, the system aborts any additional instances of the job while the current instance is running.

1. Right-click the job in the project area choose *Properties*.
2. Open the *Execution Options* tab.
3. Select the *Single instance* checkbox.
4. Click *OK*.

Related Information

[Stopping a running job](#)

24.3 Executing jobs as immediate tasks

Immediate or "on demand" tasks are initiated from the Designer. Both the Designer and Job Server must be running for the job to execute.

24.3.1 Executing a job as an immediate task

1. In the project area, select the job name.
2. Right-click and choose *Execute*.

The software prompts you to save any objects that have changes that have not been saved.

3. The next step depends on whether you selected the *Perform complete validation before job execution* checkbox in the Designer Options:
 - If you have not selected this check box, a window opens showing execution properties (debug and trace) for the job. Proceed to the next step.
 - If you have selected this check box, the software validates the job before it runs. You must correct any serious errors before the job will run. There might also be warning messages—for example, messages indicating that date values will be converted to datetime values. Correct them if you want (they will not prevent job execution) or click *OK* to continue. After the job validates, a window opens showing the execution properties (debug and trace) for the job.
4. Set the execution properties.

You can choose the Job Server that you want to process this job, datastore profiles for sources and targets if applicable, enable automatic recovery, override the default trace properties, or select global variables at runtime.

i Note

Setting execution properties here affects a temporary change for the current execution only.

5. Click [OK](#).

As the software begins execution, the execution window opens with the trace log button active.

Use the buttons at the top of the log window to display the trace log, monitor log, and error log (if there are any errors).

After the job is complete, use an RDBMS query tool to check the contents of the target table or file.

Related Information

[Designer General options \[page 37\]](#)

[Trace properties \[page 764\]](#)

[Debugging execution errors \[page 761\]](#)

[Setting a global variable value as a job property \[page 713\]](#)

[Setting a global variable value as an execution property \[page 714\]](#)

[Execution Options \[page 756\]](#)

24.3.2 Monitor tab

The [Monitor](#) tab lists the trace logs of all current or most recent executions of a job.

The traffic-light icons in the [Monitor](#) tab have the following meanings:

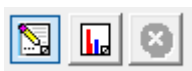
- A green light indicates that the job is running
You can right-click and select Kill Job to stop a job that is still running.
- A red light indicates that the job has stopped
You can right-click and select Properties to add a description for a specific trace log. This description is saved with the log which can be accessed later from the Log tab.
- A red cross indicates that the job encountered an error

24.3.3 Log tab

You can also select the [Log](#) tab to view a job's trace log history.

Click a trace log to open it in the workspace.

Use the trace, monitor, and error log icons (left to right at the top of the job execution window in the workspace) to view each type of available log for the date and time that the job was run.



24.4 Debugging execution errors

The following table lists tools that can help you understand execution errors:

Tool	Definition
Debug package	Creates a job debug package that includes the ATL and trace, monitor, and log files. This will help you and SAP customer support troubleshoot execution-related issues.
Error log	Displays the name of the object being executed when an error occurred and the text of the resulting error message. If the job ran against SAP data, some of the ABAP errors are also available in the error log.
Monitor log	Displays each step of each data flow in the job, the number of rows streamed through each step, and the duration of each step.
Target data	Always examine your target data to see if your job produced the results you expected.
Trace log	Itemizes the steps executed in the job and the time execution began and ended.

Related Information

[Access logs \[page 761\]](#)

[Examining trace logs \[page 763\]](#)

[Examining monitor logs \[page 763\]](#)

[Examining error logs \[page 763\]](#)

[Examining target data \[page 767\]](#)

[Creating a debug package \[page 767\]](#)

24.4.1 Access logs

To view information about the job execution, use log files generated by Data Services.

To open the trace log after job execution:

Select **Tools** > **Options** > **Designer** > **General** > **Open monitor on job execution**.

To copy log content from an open log, select one or multiple lines and use click **Ctrl** + **C**.

24.4.1.1 Accessing a log during job execution

If you set up your options to open the monitoring pane when you execute jobs, and if your Designer is running when job execution begins, the execution window opens automatically, displaying the trace log information.

Click the monitor and error log icons (middle and right icons at the top of the monitoring pane) to view the the monitor and error logs. The error log icon is available only if your job execution produces warnings or errors. The warning icon (yellow triangle with exclamation point) icon indicates that the job has warnings. The warning icon changes to an error icon (red circle with a white X on top) when there are errors in the job execution.



You can click the various icons while the job is executing to view each log. The monitoring pane stays open until you close it.

24.4.1.2 Accessing a log after the execution window has been closed

1. In the project area, click the [Log](#) tab.
2. Click a job name to view all trace, monitor, and error log files in the workspace. Or expand the job you are interested in to view the list of trace log files and click one.

Log indicators signify the following:

Job Log Indicator	Description
	Indicates that the job executed successfully on this explicitly selected Job Server.
	Indicates that the was job executed successfully by a server group. The Job Server listed executed the job.
	Indicates that the job encountered an error on this explicitly selected Job Server.
	Indicates that the job encountered an error while being executed by a server group. The Job Server listed executed the job.

3. Click the log icon for the execution of the job you are interested in. (Identify the execution from the position in sequence or datetime stamp.)
4. Use the list box to switch between log types or to view [No logs](#) or [All logs](#).

24.4.1.3 Deleting a log

Delete log files from the Project area in Designer.

Caution

Before you delete any log files make sure that you don't need the information in the log files for reporting or statistical purposes. Some log files contain important information that the software needs to create regulatory or certified reports as well as statistical tables, and some log files are overwritten when you run another job. Therefore we suggest that if you need to create regulatory, certified, or statistical data based on a specific process, you create backup copies of the log files before you delete them.

To delete log files manually from designer:

1. In the project area, click the [Log](#) tab.
2. Right-click the log you want to delete and select [Delete Log](#).

Related Information

[Administrator Guide: Server Management, Server maintenance, History file and log file retention](#)

24.4.1.4 Examining trace logs

Use the trace logs to determine where an execution failed, whether the execution steps occur in the order you expect, and which parts of the execution are the most time consuming.

24.4.1.5 Examining monitor logs

The monitor log quantifies the activities of the components of the job. It lists the time spent in a given component of a job and the number of data rows that streamed through the component.

24.4.1.6 Examining error logs

The software produces an error log for every job execution. Use the error logs to determine how an execution failed. If the execution completed without error, the error log is blank.

24.4.1.7 Disable warnings that occur during job execution

Disable warnings that are logged during runtime.

By default, Data Services logs a warning for every record it hits. This can make it hard to navigation through the log.

To minimize the number of warning you see in the error log, do the following:

1. Open the `DSConfig.txt` file, located in `%DS_COMMON_DIR%\conf`.
2. Set the `Disable_Runtime_warning` parameter, in the `AI_Engine` section, to `True`.
3. Save and close the file.

24.4.2 Trace properties

Trace properties are types of information that the software monitors during job execution, and writes to trace files.

Use the [Trace](#) tab of the [Properties](#) window to select the information that SAP Data Services monitors and writes to the trace log file during a job. The software writes trace messages to the trace log associated with the current Job Server and writes error messages to the error log associated with the current Job Server.

To enable or disable a trace property, open the [Trace](#) tab and select [Yes](#) or [No](#) in the [Value](#) list for the applicable property.

The following table contains descriptions for each trace property in alphabetical order.

Trace	Description
Access Server Communication	Writes messages exchanged between the Access Server and a service provider, including the following message types: • A registration message, which tells the Access Server that the service provider is ready. • A request to execute that the Access Server sends to the service provider. • A response from the service provider to the Access Server. • All requests from the Access Server to shut down.
Assemblers	Writes messages for Substitution Parameter and SDK Transforms: • Substitution parameters: Writes trace messages such as the substitution configuration used and the value substituted for each substitution parameter. • SDK transforms: Writes transform-specific information specified in XML format. The transform-specific information can be hierarchical. The Assemblers trace message writes the XML extracted or assembled at runtime.
Audit Data	Writes a message when auditing: • Collects a statistic at an audit point • Determines if an audit rule passes or fails

Trace	Description
Data Flow	Writes a message when the data flow starts, when the data flow successfully finishes, or when the data flow terminates due to error. This trace also reports when the bulk loader starts, any bulk loader warnings occur, and when the bulk loader successfully completes.
Debug Package	Creates a package containing the trace, monitor, and log files and the ATL for the job.
IDoc file reader	Writes a message when reading IDoc files including: • start reading the IDoc file • stop reading the IDoc file • result of the IDoc file validation
Memory Source	Writes a message for every row retrieved from the memory table.
Memory Target	Writes a message for every row inserted into the memory table.
Optimized data flow	For consulting and customer assurance use.
RFC Function	Writes a message related to RFC calls including: • start of RFC call • end of RFC call • a message for each record received from the software for the RFC call
Row	Writes a message when a transform imports or exports a row.
SAP Table Reader	Writes messages from readers of SAP system tables including: • start reading from table • stop reading from table • start of connection to SAP system where table is present • end of connection to SAP system
Scripts and Script Functions	Writes a message when the software runs a script or calls a script function. Specifically, this trace links a message when: • The script is called. Scripts can be started any level from the job level down to the data flow level. Additional (and separate) notation is made when a script is called from within another script. • A function is called by the script. • The script successfully completes.
Session	Writes a message when the job description is read from the repository, when the job is optimized, and when the job runs.

Trace	Description
SQL Functions	Writes data retrieved before SQL functions: • Every row retrieved by the named query before the SQL is submitted in the <code>key_generation</code> function • Every row retrieved by the named query before the SQL is submitted in the <code>lookup</code> function (but only if <code>PRE_LOAD_CACHE</code> is not specified). • When mail is sent using the <code>mail_to</code> function.
SQL Loaders	Writes a message when the bulk loader: • Starts • Submits a warning message • Completes successfully • Completes unsuccessfully, if you select the <i>Clean up bulk loader directory after load</i> option. Additionally, for Microsoft SQL Server and SAP ASE, writes when the SQL Server bulk loader: • Completes a successful row submission • Encounters an error This instance reports all SQL that the software submits to the target database, including: • When a <code>truncate table</code> command executes if the <i>Delete data from table before loading</i> option is selected. • Any parameters included in PRE-LOAD SQL commands • Before a batch of SQL statements is submitted • When a template table is created (and also dropped, if you turn on the <i>Drop/Create</i> option) • When a <code>delete from table</code> command executes if you turn on auto correct (Informix environment only). This trace also writes all rows that the software loads into the target.
SQL Only	Use with <i>Trace the SQL Transforms</i> option, the <i>Trace SQL Readers</i> option, or the <i>Trace SQL Loaders</i> option to stop the writing of trace messages for data sent and received from the database.
SQL Readers	Writes the SQL query block that a script, Query transform, or SQL function submits to the system. Also writes the SQL results.
SQL Transforms	Writes a message (using the <code>Table_Comparison</code> transform) about whether a row exists in the target table that corresponds to an input row from the source table. The trace message occurs before submitting the query against the target and for every row retrieved when the named query is submitted (but only if caching is not turned on).

Trace	Description
Stored Procedure	Writes a message when the software calls a stored procedure. The report includes: • When the stored procedure starts • The SQL query submitted for the stored procedure call • The value (or values) of the input parameter (or parameters) • The value (or values) of the output parameter (or parameters) • The return value (if the stored procedure is a stored function) • When the stored procedure finishes
Tables	Writes a message when a table is created or dropped. The message indicates the datastore to which the created table belongs and the SQL statement used to create the table.
Trace ABAP	Writes a message when an ABAP data flow starts or stops, and to report the ABAP job status.
Trace Parallel Execution	Writes messages describing how data in a data flow is parallel processed.
Transform	Writes a message when a transform starts, completes, or terminates.
Work Flow	Writes a message when the work flow description is read from the repository, when the work flow is optimized, when the work flow runs, and when the work flow ends.

24.4.3 Examining target data

The best measure of the success of a job is the state of the target data. Always examine your data to make sure the data movement operation produced the results you expect. Be sure that:

- Data was not converted to incompatible types or truncated.
- Data was not duplicated in the target.
- Data was not lost between updates of the target.
- Generated keys have been properly incremented.
- Updated values were handled properly.

24.4.4 Creating a debug package

Create a job debug package that includes the ATL and trace, monitor, and log files.

You can create a job debug package that includes the ATL and trace, monitor, and log files. Send the generated .zip or .tar file to SAP customer support to help you troubleshoot execution-related issues. You can generate the debug package as you execute the job either from the Execution Properties window of the Designer or in the Execution Options page of the Management Console Administrator.

1. In the Designer project area, right-click the job name and choose [Execute](#).

The software prompts you to save any objects that have changes that have not been saved.

2. In the *Execution Properties* window, select the *Trace* tab.
3. Select *Create Debug Package*, and change the value to *Yes*.
4. Click *OK*.

The software generates the zip or tar debug package file in the following location: `<DS_COMMON_DIR>\log\<Job_Server_Name>\<repository name>` on the machine where the Job Server is installed.

The debug package contains the following files: `dsconfig_flaglog.txt` (contains DSConfig flags and their corresponding values that are being used in the job, which can help you troubleshoot an environment when moving from a working to non-working environment), `DSConfig.txt`, `tracelog.txt`, `errorlog.txt`, `job.atl`, and `sysinfo_tracelog.txt`.

You can download the debug package from the Management Console Administrator. For more information, see the *SAP Data Services Management Console Guide*.

24.5 Changing Job Server options

Familiarize yourself with the more technical aspects of how the software handles data (using the *Reference Guide*) and some of its interfaces like those for adapters and SAP application.

There are many options available in the software for troubleshooting and tuning a job.

Option	Option description	Default value
Adapter Data Exchange Time-out	(For adapters) Defines the time a function call or outbound message will wait for the response from the adapter operation.	10800000 (3 hours)
Adapter Start Time-out	(For adapters) Defines the time that the Administrator or Designer will wait for a response from the Job Server that manages adapters (start/stop/status).	90000 (90 seconds)
AL_JobServerLoadBalance-Debug	Enables a Job Server to log server group information if the value is set to TRUE. Information is saved in: <code>< \$LINK_DIR> /log/<JobServerName>/server_eventlog.txt</code> .	FALSE
AL_JobServerLoadOSPolling	Sets the polling interval (in seconds) that the software uses to get status information used to calculate the load balancing index. This index is used by server groups.	60

Option	Option description	Default value
Display DI Internal Jobs	Displays the software's internal datastore CD_DS_d0cafae2 and its related jobs in the object library. The CD_DS_d0cafae2 datastore supports two internal jobs. The first calculates usage dependencies on repository tables and the second updates server group configurations. If you change your repository password, user name, or other connection information, change the default value of this option to TRUE, close and reopen the Designer, then update the CD_DS_d0cafae2 datastore configuration to match your new repository configuration. This enables the calculate usage dependency job (CD_JOBd0cafae2) and the server group job (di_job_al_mach_info) to run without a connection error.	FALSE
FTP Number of Retry	Sets the number of retries for an FTP connection that initially fails.	0
FTP Retry Interval	Sets the FTP connection retry interval in milliseconds.	1000
Global_DOP	Sets the Degree of Parallelism for all data flows run by a given Job Server. You can also set the Degree of parallelism for individual data flows from each data flow's Properties window. If a data flow's Degree of parallelism value is 0, then the Job Server will use the Global_DOP value. The Job Server will use the data flow's Degree of parallelism value if it is set to any value except zero because it overrides the Global_DOP value.	2
Ignore Reduced Msg Type	(For SAP applications) Disables IDoc reduced message type processing for all message types if the value is set to TRUE.	FALSE
Ignore Reduced Msg Type_foo	(For SAP application) Disables IDoc reduced message type processing for a specific message type (such as <foo>) if the value is set to TRUE.	FALSE
OCI Server Attach Retry	The engine calls the Oracle OCIServerAttach function each time it makes a connection to Oracle. If the engine calls this function too fast (processing parallel data flows for example), the function may fail. To correct this, increase the retry value to 5.	3
Splitter Optimization	The software might hang if you create a job in which a file source feeds into two queries. If this option is set to TRUE, the engine internally creates two source files that feed the two queries instead of a splitter that feeds the two queries.	FALSE

Option	Option description	Default value
Use Explicit Database Links	Jobs with imported database links normally will show improved performance because the software uses these links to push down processing to a database. If you set this option to FALSE, all data flows will not use linked datastores. The use of linked datastores can also be disabled from any data flow properties dialog. The data flow level option takes precedence over this Job Server level option.	TRUE
Use Domain Name	Adds a domain name to a Job Server name in the repository. This creates a fully qualified server name and allows the Designer to locate a Job Server on a different domain. If the fully qualified host name (including domain) is not stored properly, the connection between the Designer or Job Server and the repository may fail. If this problem occurs, change this option to FALSE before adding repositories.	TRUE

Related Information

[Performance Optimization Guide: Using parallel Execution, Degree of parallelism](#)
[Linked datastores \[page 186\]](#)

24.5.1 Changing option values for an individual Job Server

- Select the Job Server you want to work with by making it your default Job Server.
 - Select **Tools > Options > Designer > Environment**.
 - Select a Job Server from the *Default Job Server* section.
 - Click *OK*.
- Select **Tools > Options > Job Server > General**.
- Enter the section and key you want to use from the following list of value pairs:

Section	Key
int	AdapterDataExchangeTimeout
int	AdapterStartTimeout
AL_JobServer	AL_JobServerLoadBalanceDebug

Section	Key
AL_JobServer	AL_JobServerLoadOSPolling
string	DisplayDIInternalJobs
AL_Engine	FTPNumberOfRetry
AL_Engine	FTPRetryInterval
AL_Engine	Global_DOP
AL_Engine	IgnoreReducedMsgType
AL_Engine	IgnoreReducedMsgType_foo
AL_Engine	OCIServerAttach_Retry
AL_Engine	SPLITTER_OPTIMIZATION
AL_Engine	UseExplicitDatabaseLinks
Repository	UseDomainName

4. Enter a value.

For example, enter the following to change the default value for the number of times a Job Server will retry to make an FTP connection if it initially fails:

Option	Sample value
Section	AL_Engine
Key	FTPNumberOfRetry
Value	2

These settings will change the default value for the FTPNumberOfRetry option from zero to two.

- To save the settings and close the Options window, click [OK](#).
- Re-select a default Job Server by repeating step 1, as needed.

24.5.2 Using mapped drive names in a path

The software supports only UNC (Universal Naming Convention) paths to directories. If you set up a path to a mapped drive, the software will convert that mapped drive to its UNC equivalent.

To make sure that your mapped drive is not converted back to the UNC path, you need to add your drive names in the *Options* window in the Designer.

1. Choose ► *Tools* ► *Options* ►.
2. In the *Options* window, expand *Job Server* and then select *General*.
3. In the *Section* edit box, enter **MappedNetworkDrives**.
4. In the *Key* edit box, enter **LocalDrive1** to map to a local drive or **RemoteDrive1** to map to a remote drive.
5. In the *Value* edit box, enter a drive letter, such as **M:** for a local drive or **\\<machine_name>\<share_name>** for a remote drive.
6. Click OK to close the window.

If you want to add another mapped drive, you need to close the *Options* window and re-enter. Be sure that each entry in the *Key* edit box is a unique name.

25 Functions and Procedures

Functions and procedures are alike because they both take input values. However, they differ in what they do with the values.

- Functions take input values and produce a return value.
- Procedures take input values and perform a set of operations without returning a specific value.

Input values include the following:

- Parameters passed into a data flow
- Values from a column of data
- Variables defined inside a script

[About functions \[page 773\]](#)

Functions in SAP Data Services take input values and produce a return value.

[About procedures \[page 787\]](#)

SAP Data Services supports stored procedures, which are executable objects or named entities that you store in a database.

25.1 About functions

Functions in SAP Data Services take input values and produce a return value.

Some functions produce the same or similar values as transforms. However, functions operate on single values and transforms operate on data sets. Use functions alone or include functions in expressions. Data Services provides many built-in functions for you to use. You can also create custom functions by using Data Services scripting language and the Smart Editor.

Data Services can perform implicit data type conversions on date, time, datetime, and interval values.

Find complete details about working with functions in the *Designer Guide*.

[Functions compared with transforms \[page 774\]](#)

Some functions can produce the same or similar values as transforms. However, functions and transforms operate in a different scope.

[Operation of a function \[page 774\]](#)

The type of function determines where you can use the function. The function operation determines where you can call the function.

[Arithmetic in date functions \[page 775\]](#)

The software performs some implicit data type conversions on date, time, datetime, and interval values.

[Including functions in expressions \[page 775\]](#)

Use a function in an expression only when the function makes sense in the expression that you create.

[Custom functions \[page 778\]](#)

When none of SAP Data Services built-in functions match your needs, create a custom function.

[Built-in functions \[page 783\]](#)

Related Information

[Smart Editor and function tools \[page 647\]](#)

[Scripting Language \[page 803\]](#)

25.1.1 Functions compared with transforms

Some functions can produce the same or similar values as transforms. However, functions and transforms operate in a different scope.

- Functions operate on single values, such as values in specific columns in a data set.
- Transforms operate on data sets, creating, updating, and deleting rows of data.

SAP Data Services does not support functions that include tables as input or output parameters, except functions imported from SAP Applications.

25.1.2 Operation of a function

The type of function determines where you can use the function. The function operation determines where you can call the function.

For example, a lookup database function operates as an iterative function. The lookup function caches information about the table and columns on which it operates between function calls.

By contrast, conversion functions, such as `to_char`, operate as stateless functions. Conversion functions operate independently in each iteration.

Aggregate functions, such as `max`, require a set of values with which to operate. You cannot call the lookup function (iterative) or the `max` function (aggregate) from a script or conditional where the context does not support how these functions operate.

The function type determines where you can use a function. The following table describes each type of function and where you can call it from.

Type	Description
Aggregate	Generates a single value from a set of values. Aggregate functions, such as <code>max</code>, <code>min</code>, and <code>count</code>, use the data set specified by the expression in the Group By tab of a query. Call an aggregate function only from within a Query transform. You cannot call an aggregate function from custom functions or scripts.

Type	Description
<i>Iterative</i>	Maintains state information from one invocation to another. An iterative function, such as the lookup function, contains state information that lasts only until you execute the query in which you use the function. Call an iterative function only from within a Query transform. You cannot call an iterative function from other functions or scripts.
<i>Stateless</i>	Does not maintain state information from one invocation to the next. Use stateless functions, such as <code>to_char</code> or <code>month</code>, anywhere you can use expressions.

25.1.3 Arithmetic in date functions

The software performs some implicit data type conversions on date, time, datetime, and interval values.

Related Information

[Date arithmetic \[page 954\]](#)

25.1.4 Including functions in expressions

Use a function in an expression only when the function makes sense in the expression that you create.

You can use functions in the following objects:

- Transforms (Query , Case, SQL, Map_Operation)
- Script objects
- Conditionals
- Other custom functions

Before you use a function, ensure that the function operation makes sense in the expression you are creating.

For example:

- You cannot use the max function in a script or conditional where there is no collection of values on which to operate.
- Parameters can be output by a work flow but not by a data flow.

There are two editors that you use to add an existing function to an expression:

- Smart Editor
Embedded in other editor windows like the Script Editor, Conditional Editor, and Query Editor. Offers color-coded syntax, a right-click menu, keyboard short cuts, and a list of available variables, data type formats, and functions that you can use to define a function.

- **Function wizard**
Use to define the parameters for an existing function. The function wizard offers the most help when defining complex functions.
[Creating an expression that includes an existing function \[page 776\]](#)
Use both the Smart Editor and the function wizard to create an expression with an existing function.
[Editing an existing function call in an expression \[page 777\]](#)
There are a variety of editors to use for editing a function call such as the Smart Editor, script editor, Query transform editor, and so on.

Related Information

[Smart Editor and function tools \[page 647\]](#)

25.1.4.1 Creating an expression that includes an existing function

Use both the Smart Editor and the function wizard to create an expression with an existing function.

Ensure that the applicable object for building the expression is available in Designer.

1. Open the script, Query transform, or conditional editor as applicable.
2. Open the Smart Editor and start creating the expression.

❖ Example

For example, in the Query transform *Mapping* tab, click the *Ellipses* button to open the Smart Editor.

3. At the point where the function belongs, click the *Functions* icon in the toolbar.

The Designer opens the *Select Function* dialog box, which is the first step in the function wizard.

4. Select a category from the list under *Function categories* in the left pane to display a list of available functions in the *Function name* list in the right pane.

Data Services displays only the functions related to the object you are using. For example, the functions available for ABAP data flows are a subset of the functions available for data flows.

In some cases, it does not make sense to use a function even though the function is available. For example, you can call the SQL function in a mapping expression or a WHERE clause, but the result is a SQL statement inside a SQL statement that the software generates to execute a data flow.

5. Select a specific function in the *Function name* list.

The software displays a description of the function in the bottom pane of the dialog box.

6. Click *Next*.

7. Enter the values required by the function in the *Define Input Parameter(s)* dialog box.

The software displays only the page applicable to the function. The function wizard designs each page to help you construct the current function. For example, click in any text box to see a description of the

parameter at the bottom of the window. Use the down-arrow button to select input parameters. Use the smart editor button (...) to access the editor pane to see a larger input box.

8. When you are done constructing the function, click *Finish* to close the function wizard.

The function and the parameters appear in the Smart Editor editor pane.

Task overview: [Including functions in expressions \[page 775\]](#)

Related Information

[Editing an existing function call in an expression \[page 777\]](#)

25.1.4.2 Editing an existing function call in an expression

There are a variety of editors to use for editing a function call such as the Smart Editor, script editor, Query transform editor, and so on.

The function wizard may save you the most time when you edit the function because it parses the different pieces of the function into separate arguments. This parsing enables you to more easily identify the options to edit. However, the function wizard removes any formatting, spacing, and comments, so if your function contains these types of things, consider using another tool.

i Note

After you finish editing in the function wizard, the function text appears on one line with minimal spacing between the parameters.

The following steps use the function wizard to edit an existing function in an expression.

1. Go to the expression that contains the function call that you want to change.
2. Right-click on the text of the function call and select *Edit Function*.
The function wizard opens.
3. Make changes to the function call, and then click *Finish*.
The function wizard closes and you can see the results of your function call.

Task overview: [Including functions in expressions \[page 775\]](#)

Related Information

[Creating an expression that includes an existing function \[page 776\]](#)

25.1.5 Custom functions

When none of SAP Data Services built-in functions match your needs, create a custom function.

After you create a custom function, you save it so it is available to use again. Data Services stores custom function in the *Custom Functions* tab of the object library under the *Custom Functions* node. When you create a validation custom function, Data Services saves it under the *Validation Functions* category in the *Custom Functions* tab.

To use a function that you created, drag and drop it from the Custom Functions tab to a specific area in the open object in the workspace. For example, drag a custom function from the *Custom Functions* tab to the *Mapping* tab of the Query transform.

Create validation functions in Data Services, or import them from SAP Information Steward.

i Note

You cannot edit validation functions that you import from Information Steward.

Validation functions that you create in Data Services are reusable and you can edit them.

Consider the following guidelines when you create your own functions:

- Functions can call other functions.
- Functions cannot call themselves.
- Functions cannot participate in a cycle of recursive calls. For example, function A cannot call function B, which calls function A.
- Functions return a value.
- Functions can have parameters for input, output, or both. However, data flows cannot pass function output parameters or input/output type parameters.

Before creating a custom function, ensure that you know the input, output, return values, and data types. The software predefines the return value as `Return`.

[Create a custom function \[page 779\]](#)

Create custom functions for validation or when there is no built-in function for your purposes.

[Creating custom functions from the Tools menu \[page 779\]](#)

In the following steps, we use the *Tools* menu to create a custom function. You can also create new custom functions from the *Custom Functions* tab in object explorer.

[Editing an existing function \[page 781\]](#)

Open existing custom functions to edit them.

[Replicating a custom function \[page 782\]](#)

You can replicate an existing custom function when you want to create a new custom function that is similar.

[Deleting a custom function \[page 782\]](#)

Delete custom functions when you no longer need them.

Related Information

25.1.5.1 Create a custom function

Create custom functions for validation or when there is no built-in function for your purposes.

A custom function consists of return values, variables, and parameters. Create custom functions in the Smart Editor. Begin to create a custom function in one of two ways:

- From the *Custom Functions* tab in the object library.
- From the *Tools* menu.

Each time that you add a return, variable, or parameter, the Smart Editor uses a default naming convention. For example, create a new variable and the Smart Editor names it "\$NewVariable1." Change the name if you want. Make sure that you include the dollar sign in front of the name that you enter.

When you accept the default name, the ending number increases by one for each subsequent variable. So when you create a second new variable, the default name is "\$NewVariable2," and so on.

When you have finished configuring the various parts of your function, start to build it in the Smart Editor. Use the Validate feature to check your syntax and to ensure that the function is correct.

The variables and parameters that you create for a custom function are local to each function. Therefore, you can't access them in the *Variables and Parameters* dialog box that opens when you select  *Tools*  *Variables* . If you edit the custom function, you can view the variables and parameters in the smart editor library, in the *Variables* tab.

Parent topic: [Custom functions \[page 778\]](#)

Related Information

[Creating custom functions from the Tools menu \[page 779\]](#)

[Editing an existing function \[page 781\]](#)

[Replicating a custom function \[page 782\]](#)

[Deleting a custom function \[page 782\]](#)

25.1.5.2 Creating custom functions from the Tools menu

In the following steps, we use the *Tools* menu to create a custom function. You can also create new custom functions from the *Custom Functions* tab in object explorer.

1. In Designer, choose  *Tools* .

The *Custom Functions* dialog box opens.

2. Right-click in a blank area of the upper pane and select [New](#).

The software adds a line in the upper pane with a text box that says “New Function.”

3. Replace “New Function” with the name of the function.
4. Select [Enable parallel execution](#) if applicable.
5. Select [Validation function](#) to create a validation function.
6. Enter a description in the [Description](#) area in the bottom pane of the [Custom Functions](#) dialog box.
7. Click [Next](#).

The Smart Editor opens. In the Smart Editor, you define the return type, create a parameter list, and create any local variables to use in the function.

8. Open the [Variables](#) tab, expand [Parameters](#), right-click [Return](#) and select [Properties](#)

The [Return Value Properties](#) dialog box opens. By default, the software sets the [Name](#) to [Return](#) and the [Data type](#) to [int](#). You cannot change the [Name](#) field.

9. To change the data type, click the [Data type](#) dropdown arrow and select a different data type.
10. Specify other settings related to the data type you chose. For example, choose [varchar](#) and the [Length](#) field appears. Choose [decimal](#) and the [Precision](#) and [Scale](#) fields appear.
11. When you are finished, click [OK](#) to close the [Return Value Properties](#) dialog box.
12. In the [Variables](#) tab of the Smart Editor, right-click [Parameters](#) and choose [Insert](#).

The [New Parameter](#) dialog box opens.

13. Enter a parameter name or accept the default name.

Parameter names begin with a dollar sign (\$). The default is `<$NewParameter1>` with the ending number counting up for each subsequent parameter that you create for this function.

14. Define other properties by choosing a [Data type](#) and a [Parameter type](#) from the dropdown arrows.

Note

Data flows cannot pass variable parameters of type [Output](#) and [Input/Output](#).

15. Click [OK](#).

Repeat the parameter-related steps for each parameter required in your function. After you add one parameter, right-click an existing parameter name and select [Insert Above](#) or [Insert Below](#).

Use the right-click menu to create, delete, or edit variables and parameters.

16. To define variables used by the function, but not passed outside the function, right-click [Local](#) in the [Variables](#) tab and choose [Insert](#).

The [New Variable](#) dialog box opens.

17. Accept the default variable name or enter a different name.

Variable names begin with a dollar sign (\$). The default is `<$NewVariable1>` with the ending number counting up for each subsequent variable that you create for this function.

18. Choose a data type in the [Variable Properties](#) dialog box and click [OK](#).

Repeat the variable-related steps for each variable that you require in your function.

19. Create your function using the pieces you just created and the Smart Editor tools.
20. Click the [Validate](#) icon to validate your function.



If your function contains syntax errors, the software lists them in an embedded pane below the editor.

21. To see where the error occurs in the text, double-click the error text.

The smart editor redraws to show the location of the error.

22. When your function is valid, click *OK* to save the function, and the variables and parameters inside.

Task overview: [Custom functions \[page 778\]](#)

Related Information

[Create a custom function \[page 779\]](#)

[Editing an existing function \[page 781\]](#)

[Replicating a custom function \[page 782\]](#)

[Deleting a custom function \[page 782\]](#)

[Smart Editor and function tools \[page 647\]](#)

25.1.5.3 Editing an existing function

Open existing custom functions to edit them.

1. In Designer, choose **Tools** > *Custom Functions* >

The *Custom Functions* dialog box opens.

2. select the function to edit and click *Next*.

The Smart Editor opens with the function displayed in the edit pane and the variables and parameters that exist for this function shown under the *Variables* tab.

Alternatively:

- Go to the *Custom Functions* tab in the object library and double-click the function that you want to edit.
- Go to the *Custom Functions* tab in the object library, right-click the function, and select *Edit*.

Task overview: [Custom functions \[page 778\]](#)

Related Information

[Create a custom function \[page 779\]](#)

[Creating custom functions from the Tools menu \[page 779\]](#)

[Replicating a custom function \[page 782\]](#)

[Deleting a custom function \[page 782\]](#)

25.1.5.4 Replicating a custom function

You can replicate an existing custom function when you want to create a new custom function that is similar.

1. From the *Custom Functions* tab in object library, right-click the applicable custom function and select *Replicate*.

The *Custom Function* editor opens with only the *Function name* field, the *Enable parallel execution* and *Validation function* checkboxes enabled.

2. Enter a new name for the function.

The name must be valid and different from the original name.

3. Click *Finish*.

The new custom function appears in the object library. Edit the function if applicable.

Task overview: [Custom functions \[page 778\]](#)

Related Information

[Create a custom function \[page 779\]](#)

[Creating custom functions from the Tools menu \[page 779\]](#)

[Editing an existing function \[page 781\]](#)

[Deleting a custom function \[page 782\]](#)

25.1.5.5 Deleting a custom function

Delete custom functions when you no longer need them.

Either before or after you delete a custom function, remove references to the function from expressions, scripts, conditionals, queries, and other custom functions.

1. In Designer, open the *Custom Transforms* tab in the object library.
2. Right-click the applicable custom function and select *View Where Used*.

The *Output* dialog box opens with the *Information* tab opened showing the dataflows that use the custom function. Either delete the references now or do it after you delete the function.

3. Right-click the function to be deleted and select *Delete*.
4. Select *Yes* to the message affirming the delete.

Task overview: [Custom functions \[page 778\]](#)

Related Information

[Create a custom function \[page 779\]](#)

[Creating custom functions from the Tools menu \[page 779\]](#)

[Editing an existing function \[page 781\]](#)

[Replicating a custom function \[page 782\]](#)

25.1.6 Built-in functions

SAP Data Services provides a set of built-in functions.

In the software, database and application functions, custom functions, and most built-in functions can be executed in parallel within the transforms in which they are used.

You can run each resource-intensive functions, such as `lookup_ext` and `count_distinct`, as a separate sub data flow that uses separate resources (memory and computer) from each other.

[Database and application functions \[page 783\]](#)

[Defining an `sap\_openhub\_processchain\_execute` function \[page 784\]](#)

Use the steps to define the `sap_openhub_processchain_execute` function as a guide for defining the function in your environment.

[Defining a `search\_replace` function \[page 785\]](#)

Define a `search_replace` function using the function wizard.

[Defining and enabling the `smtp\_to` function \[page 786\]](#)

Related Information

[Descriptions of built-in functions](#)

[DOP and functions](#)

[Distributed data flow execution \[page 295\]](#)

[Catch error functions](#)

25.1.6.1 Database and application functions

You can import the metadata for database and application functions and use them in Data Services applications. You can also import stored functions and procedures.

At run time, Data Services passes the appropriate information to the database or application from which the function was imported.

The metadata for a function includes the input and output parameters and their data types. If there are restrictions on data passed to the function, such as requiring uppercase values or limiting data to a specific range, you must enforce these restrictions in the input. You can either test the data before extraction or include logic in the data flow that calls the function.

Parent topic: [Built-in functions \[page 783\]](#)

Related Information

[Defining an sap_openhub_processchain_execute function \[page 784\]](#)

[Defining a search_replace function \[page 785\]](#)

[Defining and enabling the smtp_to function \[page 786\]](#)

[About procedures \[page 787\]](#)

[Designer Guide: Imported stored function and procedure information \[page 110\]](#)

25.1.6.2 Defining an sap_openhub_processchain_execute function

Use the steps to define the sap_openhub_processchain_execute function as a guide for defining the function in your environment.

1. Right-click an empty area of the *Functions* tab in the object editor of Designer.
The *Smart Editor* dialog box opens.
2. Click the *Function Wizard* icon.
The *Select Function* dialog box opens.
3. Select *SAP Functions* from the *Function categories* list
4. Select *sap_openhub_processchain_execute* from the *Function name* list. Click *Next*.
The *Define Parameters* dialog box opens.
5. Click the dropdown arrow next to *Datastores* and select an SAP BW Source datastore name.
You can also select a substitution variable from the list. If you type in a datastore name, put single quotes around the name.
6. Click the dropdown arrow next to *Open Hub table* and select the name of a Open Hub table.
Only the names of the imported Open Hub Tables appear in this list.
7. Click the dropdown arrow next to *Process chain* and select the name of a Process Chain.
8. Required only when you want to use the variable. For *Log ID*, specify a variable that results in the BW log ID for the process chain after the function executes.
9. Required only when you want to use the variable. For *Return text*, specify a variable that returns the description of the status after the function executes.
10. Click *Finish*.

Task overview: [Built-in functions \[page 783\]](#)

Related Information

[Database and application functions \[page 783\]](#)

[Defining a search_replace function \[page 785\]](#)

[Defining and enabling the smtp_to function \[page 786\]](#)

25.1.6.3 Defining a search_replace function

Define a search_replace function using the function wizard.

1. In the Query editor, right-click the output schema area and select *New Function Call* from the dropdown menu.
2. Click *String Functions* from the Function categories list.
3. Click *search_replace* from the Function name list.
4. Click *Next*.
The *Search_replace - Select Parameters* dialog box opens.
5. Select an option for *Search type* based on the descriptions in the following table.

Search type	Description
Full string	Matches and replaces the entire contents of the input expression.
Word	Replaces only the word that matches the search value. The software does not modify any unmatched data. A word is a set of characters set apart by whitespace.
Substring	Replaces only a specific substring of characters found next to or between other characters in the input expression. The software does not modify any unmatched data. For a substring search, the software orders the search values by length from largest to smallest, and it substitutes each substring only once.

6. To ignore casing differences between the specified search values and the values from the input expressions, uncheck *Case sensitive*.
7. Define the input expressions to search. Select a column name from the input schema or use any expression that uses one or more input columns.
8. Define the output column names and lengths for the replacement values. Rename the output column by selecting the value and pressing **F2**.
9. Configure the search and replace table.
 - For an internal table, select *Internal* and specify search values and corresponding replacement values. The software stores the values for an internal table in the metadata repository as part of the function definition.
 - For an external relational table, select *External*. Specify a valid datastore or file format as the Source, and select the columns containing the search and replacement values.
 - For custom SQL, select *Custom SQL*. Choose the source datastore, and specify the columns containing the search and replacement values. Define the custom SQL text to run.
10. Specify the default replacement value for any rows that do not match the defined search values. If the replacement value is a fixed string, enclose it in single quotes.

i Note

The default replacement value applies only for a full string search, not for word or substring searches.

If you do not define a default replacement value, the software preserves the original values. The software does not make replacements for expressions that it does not find.

11. If you want to force the search_replace function to execute in a separate process, check [Run as a separate process](#). The software always loads the search and replace table into memory, so for large tables, running this function as a separate process can improve performance.

Example: Case-sensitive substring search and replace

In this example, the software replaces the longest search value first. As a result, the substring AUS does not become AUSA, but is correctly replaced with Australia.

Internal Search and Replace table:

Search value	Replace value
US	USA
AUS	Australia

Search and Replace results:

Value on input	Value on output
California, US	California, USA
Melbourne, AUS	Melbourne, Australia

Task overview: [Built-in functions \[page 783\]](#)

Related Information

[Database and application functions \[page 783\]](#)

[Defining an sap_openhub_processchain_execute function \[page 784\]](#)

[Defining and enabling the smtp_to function \[page 786\]](#)

[Run as separate process](#)

25.1.6.4 Defining and enabling the smtp_to function

1. Right-click in an empty area of the [Custom Functions](#) tab in object library and select [New](#).

The Custom Function dialog box opens.

2. Enter a function name and description and click [Next](#).

The Smart Editor opens.

3. Click the Function Wizard icon.

The [Select Function](#) dialog box opens.

4. Click [System Functions](#) under Function categories.
5. Click [smtp_to](#) under Function name.
6. Click [Next](#).

The [Define Input Parameters](#) dialog box opens.

7. Click in each option to see a description of what to enter for the option. Complete the following options:
 - Recipients list
 - Subject
 - Message
 - TraceLog lines number
 - ErrorLog lines number.
8. Click [Finish](#).
9. Open the Data Services Server Manager:
10. Enter the SMTP server settings:
 - [Server name](#): Type the name or IP address of the SMTP server (for example mail.company.com)
 - [Sender](#): Type the e-mail address that will appear in the [From](#) field of the e-mail.
11. Click [Apply](#).

Task overview: [Built-in functions \[page 783\]](#)

Related Information

[Database and application functions \[page 783\]](#)

[Defining an sap_openhub_processchain_execute function \[page 784\]](#)

[Defining a search_replace function \[page 785\]](#)

25.2 About procedures

SAP Data Services supports stored procedures, which are executable objects or named entities that you store in a database.

A stored procedure is one or more precompiled SQL statements. By calling a stored procedure from within Data Services, you can call business logic that you have already coded. The ability to use your existing business logic enables you to quickly and conveniently develop data extraction and data management tasks. You can also use stored procedures for the following purposes:

- Maintain business logic rules and provide a single point of control to ensure rules are accurate and enforced
- Reduce network overhead with client/server applications. The software stores procedures on the database server and retains compiled execution plans for procedures in the data dictionary.

Data Services supports using stored procedures for the following databases:

- DB2
- ODBC
- Oracle
- Microsoft SQL Server
- SAP HANA
- SAP SQL Anywhere
- SAP ASE
- SAP Sybase IQ
- Teradata

Call stored procedures from the jobs you create and run in Data Services.

Stored procedures must exist in the database before you can use the procedure in Data Services. Create a stored procedure in a database using the client tools provided with the database, such as Oracle SQL *Plus. After it is created, users who have execution permissions for procedures can call the stored procedure. After you import a stored procedure into Data Services, use stored procedures like functions in Data Services jobs.

Stored procedures include parameters. Each parameter has the following elements:

- Name
- Data type
- Mode such as `IN`, `INOUT`, or `OUT`

A stored procedure can use a `NULL` or default parameter value for its input and can produce more than one output parameter value.

For complete information about working with procedures, see the *Designer Guide*.

[Before you use procedures \[page 788\]](#)

Before you use stored procedures, ensure that you meet the specific Data Services requirements.

[Creating stored procedures in a database \[page 789\]](#)

[Importing metadata for stored procedures \[page 794\]](#)

[Structure of a stored procedure \[page 795\]](#)

[Calling stored procedures \[page 796\]](#)

[Checking execution status \[page 802\]](#)

25.2.1 Before you use procedures

Before you use stored procedures, ensure that you meet the specific Data Services requirements.

Ensure that you meet these requirements before you use stored procedures with Data Services.

- Ensure that the client and server versions match.

- Use only user-defined stored procedures. Data Services does not support stored procedures provided by a database system.

User-defined stored procedures or stored functions must meet the following additional requirements:

- Ensure that your procedure or stored function returns Data Services-supported data types. A return type, for example is the result value.
- Do not use the long data type in stored procedures.
- Create a unique name for the stored procedure. The name is a combination of the datastore name, owner name, and procedure name.
- When you have multiple procedures or functions with the same name in a package, know that Data Services imports only the first procedure or function with a particular name.

❖ Example

When you use Oracle, and you have multiple procedures or functions with the same name in a package, to be able to use all of them, provide a unique name for each one.

- Data Services does not support overloading. You can overload a procedure when multiple versions exist in a particular package.

Data Services validates the user name specified in the datastore that you used to import the stored procedure. This validation ensures that the data flow that calls a stored procedure enforces restrictions associated with that procedure, such as:

- Data input restrictions
- Execution privileges

:

To use stored procedures with Teradata

- Enable the use of stored procedures within the Teradata ODBC driver options.
- Call the stored procedures in the same session mode in which it is enabled.

In Windows, when you configure the Teradata ODBC driver, ensure that you make the following settings:

- In the *Teradata ODBC Driver Options* dialog box, select the *Disable CALL to EXEC Conversion* checkbox.
- In the same dialog box, set the session mode as desired in the *Session Mode* dropdown list.

For UNIX platforms, edit the file `odbc.ini` to set the option **Disable CALL To EXEC Conversion=YES**. Set the session mode option as desired, and save the file.

Ensure that you call the stored procedures in the same session.

For more information about Teradata ODBC driver configuration, see your Teradata ODBC driver configuration documentation.

25.2.2 Creating stored procedures in a database

This section provides:

- An example scenario for using a stored procedure
- Tips for creating stored procedures on each database
- Example syntax for DB2, Oracle, SAP HANA, SQL Server, and SAP ASE databases based on the example scenario

Since syntax varies between databases, the SQL statements needed to create a stored procedure for any scenario vary. The client tools used to pass the SQL statements to the database server also vary. Refer to your database documentation for more detailed descriptions and examples regarding how to create stored procedures.

In the following example, the source database has a stored procedure `Get_emp_rec` that retrieves an employee's name and hire date from the `Employee` table using a given employee number. The stored procedure takes an input parameter `Emp_number` and returns two values `Emp_name` and `Hire_date` via output parameters.

The schema of the database table `Employee` is as follows:

Name	Null?	Type
empno	Not Null	Integer
ename	Null	Varchar(20)
job	Null	Varchar(9)
mgr	Null	Integer
hiredate	Null	Date
sal	Null	Decimal(7,2)
comm	Null	Decimal(7,2)
deptno	Null	Integer

[Creating stored procedures in DB2 \[page 790\]](#)

[Creating stored procedures in Oracle \[page 791\]](#)

[Creating stored procedures in SAP HANA \[page 792\]](#)

SAP Data Services supports SAP HANA stored procedures with zero, one, or more output parameters.

[Creating stored procedures in MS SQL Server or SAP ASE \[page 793\]](#)

25.2.2.1 Creating stored procedures in DB2

In DB2, a parameter can be `IN`, `OUT` or `INOUT`. Any SQL data type can be used as a data type of a parameter. User defined data types are not supported.

Example: The DB2 syntax for the stored procedure

```
CREATE PROCEDURE GET_NAME_USING_ID (IN NID INTEGER, OUT outVar varchar(20))  
language SQL  
reads sql data  
BEGIN  
select first_name into outVar FROM CONTACT where id = NID;  
END
```

Reads sql data and language SQL are two of the many options that DB2 stored procedures support. Refer to the DB2 documentation for the DB2 stored procedure options and their meaning.

Parent topic: [Creating stored procedures in a database \[page 789\]](#)

Related Information

[Creating stored procedures in Oracle \[page 791\]](#)

[Creating stored procedures in SAP HANA \[page 792\]](#)

[Creating stored procedures in MS SQL Server or SAP ASE \[page 793\]](#)

[Creating stored procedures in a database \[page 789\]](#)

25.2.2.2 Creating stored procedures in Oracle

Oracle supports both stored procedures and stored functions. Any stored procedure that returns a value is called a stored function. Oracle is the only database to allow return values with data types other than an integer.

In Oracle, stored procedures are created using the `CREATE [OR REPLACE] PROCEDURE` statement, and stored functions are created using the `CREATE [OR REPLACE] FUNCTION` statement. The `OR REPLACE` option allows you to override an existing definition of a procedure or function with the same name.

Oracle supports `IN`, `OUT`, and `INOUT` parameters. An `INOUT` parameter allows you to pass in a parameter, modify it, and return the modified value. You can use the `DEFAULT` keyword or the assignment operator to give an `IN` parameter a default value. When an `IN` parameter has a default value, you can omit the parameter from the argument list when you call the procedure. If you do specify an argument value in the call, the specified value overrides the default value. `INOUT` and `OUT` parameters must be specified.

Example: Syntax

The Oracle syntax for the stored procedure:

```
CREATE OR REPLACE PROCEDURE  
get_emp_rec (Emp_Number IN NUMBER,  
Emp_Name OUT VARCHAR, Emp_Hiredate OUT DATE) AS
```

```

BEGIN
SELECT ename, hiredate
INTO Emp_Name, Emp_Hiredate
FROM Employee
WHERE empno = Emp_Number;
END;
/

```

In this example, the parameters are declared as IN and OUT, but Oracle also supports an INOUT parameter type, which allows you to pass in a parameter, modify it, and return the modified value. INOUT parameters can take default values. This means that the parameter can be omitted from the actual parameter list when you call the procedure. INOUT and OUT parameters must be specified.

An Oracle package is an encapsulated collection of related program objects (e.g., procedures, functions, variables, constants, cursors, and exceptions) stored together in the database. Data Services allows you to import procedures or functions created within packages and use them the same way as top-level procedures or functions.

Data Services does not support overloading of procedures or functions. Overloading a procedure means creating multiple procedures with the same name in the same package, each taking arguments of a different number or data type. If you have multiple procedures or functions with the same name in the same package (and you want to use all of them), you need to rename the procedures or functions so that they all have distinct names. Otherwise, Data Services will only import the first procedure or function by that name.

Parent topic: [Creating stored procedures in a database \[page 789\]](#)

Related Information

[Creating stored procedures in DB2 \[page 790\]](#)

[Creating stored procedures in SAP HANA \[page 792\]](#)

[Creating stored procedures in MS SQL Server or SAP ASE \[page 793\]](#)

[Creating stored procedures in a database \[page 789\]](#)

25.2.2.3 Creating stored procedures in SAP HANA

SAP Data Services supports SAP HANA stored procedures with zero, one, or more output parameters.

Data Services supports scalar data types for input and output parameters. Data Services does not support table data types. If you try to import a procedure with table data type, the software issues an error. Data Services does not support data types such as binary, blob, clob, nclob, or varbinary for SAP HANA procedure parameters.

Procedures can be called from a script or from a Query transform as a new function call.

❖ Example

Syntax

The SAP HANA syntax for the stored procedure:

```
CREATE PROCEDURE GET_EMP_REC (IN EMP_NUMBER INTEGER, OUT EMP_NAME
VARCHAR(20), OUT EMP_HIREDATE DATE) AS
BEGIN
    SELECT ENAME, HIREDATE
        INTO EMP_NAME, EMP_HIREDATE
    FROM EMPLOYEE
    WHERE EMPNO = EMP_NUMBER;
END;
```

Limitations

SAP HANA provides limited support of user-defined functions that can return one or several scalar values. These user-defined functions are usually written in L. If you use user-defined functions, limit them to the projection list and the GROUP BY clause of an aggregation query on top of an OLAP cube or a column table. These functions are not supported by Data Services.

SAP HANA procedures cannot be called from a WHERE clause.

For more information about creating stored procedures in a database, see the Functions and Procedures section in the *Designer Guide*.

Parent topic: [Creating stored procedures in a database \[page 789\]](#)

Related Information

[Creating stored procedures in DB2 \[page 790\]](#)

[Creating stored procedures in Oracle \[page 791\]](#)

[Creating stored procedures in MS SQL Server or SAP ASE \[page 793\]](#)

25.2.2.4 Creating stored procedures in MS SQL Server or SAP ASE

In Microsoft SQL Server and SAP ASE, a stored procedure may have a group number (between 1 and 32767), which is an optional integer used to group procedures of the same name so they can be dropped together with a single `DROP PROCEDURE` statement.

The first procedure you create with a name automatically is assigned group number 1. While you can execute this procedure without specifying the group number, other procedures with the same procedure name must be called with the group number. For example, `execute orderproc;2` and so on. Data Services imports the procedure with the group number 1, i.e., the first procedure, unless you specify the name including the group number, such as `orderproc;2`.

A parameter can be either an input parameter (default) or an output parameter. A parameter can not be both input and output. In the following sample, the first parameter @Emp_Number is an input parameter and the other two parameters, @Emp_Name and @Emp_Hiredate, are output parameters. Output parameters are specified using the keyword OUTPUT. Input parameters can take default values (database defaults). If a default value is defined, the procedure can be executed without specifying a value for that parameter. You specify a parameter name with "@" as the first character.

An optional integer return value is automatically added to each stored procedure to store its return status. In the following example, if the SELECT statement is successful, "0" is returned as the return value. The parameter name of the return value is RETURN_VALUE. Microsoft SQL Server and SAP ASE stored procedures can be used like Oracle stored functions in Data Services because they include this return value.

A user-defined function is a new feature of SQL Server 2000. Data Services supports user-defined stored procedures, but not user-defined functions at this time.

Example: The MS SQL Server and SAP ASE syntax

```
CREATE PROCEDURE GET_EMP_REC @Emp_Number int, @Emp_Name varchar(20) OUTPUT,  
@Emp_Hiredate datetime OUTPUT AS  
SELECT @Emp_Name = ename, @Emp_Hiredate = hiredate  
FROM Employee  
WHERE empno = @Emp_Number  
RETURN 0
```

Parent topic: [Creating stored procedures in a database \[page 789\]](#)

Related Information

[Creating stored procedures in DB2 \[page 790\]](#)

[Creating stored procedures in Oracle \[page 791\]](#)

[Creating stored procedures in SAP HANA \[page 792\]](#)

[Creating stored procedures in a database \[page 789\]](#)

25.2.3 Importing metadata for stored procedures

Use the Import by Name command to import stored procedures into Data Services. You must know the name of the stored procedure, stored function, or package you want to import. Use the name of the stored procedure, the package name only, or, if you are using a Microsoft SQL Server or SAP ASE database, use the name of the stored procedure and the group number, such as `orderproc;2`.

Stored procedures are created and structured very much like functions. Imported stored procedures appear on the Datastores tab of the object library. Each is listed in the Functions category of the database datastore from which the procedure was imported. It is displayed with the following syntax:

```
<datastore>.<owner>.<proc_name>.
```

If a stored procedure was defined within an Oracle package, the name of the procedure is displayed as

```
<datastore>.<owner>.<package>.<proc_name>.
```

To view the signature of the procedure, right-click the imported stored procedure and select Open (or simply double-click the procedure).

- A stored procedure has a signature that consists of its defined parameters including two output parameters added by Data Services during import, `AL_SP_RETCODE` and `AL_SP_ERRMSG`.
- If the database you are using provides a return code for a stored procedure, it is displayed in the signature as `AL_SP_RETURN`.
- Each parameter has a name, a data type, and a mode (`IN`, `INOUT`, or `OUT`). The mode is indicated by icons shown before the parameter name.
- Oracle stored functions have a signature that consists of parameters and a return data type. When you call a function, the return value is named `AL_SP_RETURN`.

i Note

If you change the signature of a stored procedure, for example you add a new parameter, re-import the metadata for the procedure from the database.

Related Information

[Checking execution status \[page 802\]](#)

25.2.4 Structure of a stored procedure

Imported information for a stored procedure includes:

- Name and owner
- Return data type if the stored procedure has a return value or code
- Parameters

For each parameter, Data Services imports the following information:

- Name
- Mode (`IN`, `OUT`, or `INOUT`)
- data type (length, precision and scale if applicable)

i Note

In an Oracle stored procedure declaration, it is illegal to constrain `char` and `varchar2` parameters with a length, or to constrain number parameters with a precision and/or scale. These constraints are taken

from the procedure arguments. Therefore, the length of a char or varchar and the precision and scale of a number data type parameter are not stored in the data dictionary.

However, Data Services does not allow the length of a varchar data type or the precision of a number data type to be 0. (Data Services converts char data types in Oracle to varchar.) When importing a stored procedure, Data Services fills in the default length for a varchar or the precision and scale for a number parameter. For varchar, Data Services sets the length to 4000. For number, Data Services sets precision and scale to the values set in the Advanced section of the Create New Datastore window. At runtime, the length of varchar and the precision and scale of number come from the procedure arguments.

- Position within the parameter list
- Nullable (whether a default value exists)

Data Services omits a parameter if any of the following data type conditions is true:

- Is not a Data Services supported data type or is `long`
- Data type is missing in the database dictionary
- Is a composite data type (e.g., table, record, or cursor)

After Data Services imports a stored procedure, the Designer reports warnings about any omitted parameters.

i Note

Even if a stored procedure has an omitted parameter, the procedure is callable if the parameter has a default value defined. If the omitted parameter does not have a default value, the procedure is not callable and Data Services throws a runtime error.

If a stored procedure has a return value, its result type (the data type of the result value) must be a Data Services supported database server data type. Otherwise, the procedure cannot be imported and used within Data Services applications.

For each stored procedure, Data Services automatically adds two optional output parameters, `AL_SP_RETCODE` and `AL_SP_ERRMSG`. These parameters allow you to check the execution status of the stored procedure.

Related Information

[Checking execution status \[page 802\]](#)

[Data Types](#)

25.2.5 Calling stored procedures

You can call a stored procedure just as you call custom functions in Data Services.

- When you call a stored procedure inside a script, conditional or custom function, Data Services evaluates the call as an independent SQL statement.
- When you call a stored procedure in a query's column mapping or WHERE clause, (Oracle only) Data Services can combine the SQL for the stored procedure with the SQL for the query.

- When you call a stored procedure in a SQL transform, you must use proper syntax. Data Services does not validate SQL in a SQL transform.
- A stored procedure can be used within an expression. For example:

```
ora_ds.acta_user.get_emp_sal (EMPLOYEE.NAME) + 10
ora_ds.acta_user.get_emp_sal (EMPLOYEE.NAME) < 100
```

[Stored procedure input parameters \[page 797\]](#)

[Calling stored procedures from scripts \[page 798\]](#)

[Calling stored procedures from queries \[page 799\]](#)

[Without the function wizard \[page 800\]](#)

Related Information

[About procedures \[page 787\]](#)

[Stored procedure input parameters \[page 797\]](#)

[Calling stored procedures from scripts \[page 798\]](#)

[Calling stored procedures from queries \[page 799\]](#)

[Without the function wizard \[page 800\]](#)

25.2.5.1 Stored procedure input parameters

Stored procedures can require input values for some parameters but not others. You must determine the requirements of the stored procedure and prepare the appropriate inputs. In Data Services, input parameters can be constants or expressions. In addition, Data Services supports the following:

- If an input parameter has a default value, you can omit the parameter in the call by leaving the parameter unspecified in the function wizard. The database will use the parameter's default value when evaluating the stored procedure call.
- All OUT or INOUT parameters must be specified except the two Data Services adds at import: AL_SP_RETCODE and AL_SP_ERRMSG.
- All OUT or INOUT parameters must be variables (except in a query output schema). A variable must be declared before use.
- If a stored procedure returns a value (such as an Oracle function or a MS SQL Server or SAP ASE stored procedure), you can use the procedure in an expression.
- You can call a stored procedure as a step inside an output schema of a query, script, or custom function. For example:

```
As an Oracle user you create two Oracle stored procedures: GET_EMP_SALARY and
SET_EMP_SALARY
CREATE OR REPLACE FUNCTION GET_EMP_SALARY(Emp_Name IN VARCHAR)
RETURN NUMBER AS
Return_value NUMBER;
BEGIN
SELECT sal
INTO Return_value
```

```

FROM Employee
WHERE ename = Emp_Name;
RETURN Return_value;
END;
CREATE OR REPLACE PROCEDURE SET_EMP_SALARY(Emp_Name IN VARCHAR,Emp_Sal IN
NUMBER) AS
BEGIN
UPDATE Employee
SET sal = Emp_Sal
WHERE ename = Emp_Name;
END;

```

Parent topic: [Calling stored procedures \[page 796\]](#)

Related Information

[Calling stored procedures from scripts \[page 798\]](#)

[Calling stored procedures from queries \[page 799\]](#)

[Without the function wizard \[page 800\]](#)

[About procedures \[page 787\]](#)

[Calling stored procedures \[page 796\]](#)

25.2.5.2 Calling stored procedures from scripts

After importing the stored procedures into an Oracle datastore, you can use the stored procedures in a script. For example:

```

$new_sal = ora_ds.bodi_user.get_emp_salary('MARTIN') + 100;
ora_ds.bodi_user.set_emp_salary('MARTIN', $new_sal);

```

Parent topic: [Calling stored procedures \[page 796\]](#)

Related Information

[Stored procedure input parameters \[page 797\]](#)

[Calling stored procedures from queries \[page 799\]](#)

[Without the function wizard \[page 800\]](#)

[About procedures \[page 787\]](#)

[Calling stored procedures \[page 796\]](#)

25.2.5.3 Calling stored procedures from queries

You can call stored procedures from Query transforms in the mapping and WHERE clauses, or in a Query transform output schema.

Parent topic: [Calling stored procedures \[page 796\]](#)

Related Information

[Stored procedure input parameters \[page 797\]](#)

[Calling stored procedures from scripts \[page 798\]](#)

[Without the function wizard \[page 800\]](#)

[About procedures \[page 787\]](#)

[Calling stored procedures \[page 796\]](#)

[Importing metadata for stored procedures \[page 794\]](#)

25.2.5.3.1 Mapping and WHERE clauses

In a query's column mapping or WHERE clause, you can use a stored procedure in an expression, subject to the same constraints as other expressions. However, when stored procedures are used in a query's column mapping or WHERE clause, SAP Data Services can combine the SQL statement for the stored procedure call with the SQL statement for the query and submit one request to the database.

For the software to combine the SQL statements, the stored procedure must meet the following conditions:

- It must be from an Oracle database.
- It must be a stored function.
- All its parameters must be IN parameters; none can be an OUT or INOUT parameter.
- All parameter values must be specified. You cannot leave a parameter unspecified, such as to use the default value.
- All parameter values must be constants or expressions with only constants or database table column names; none can have functions or operations.
- The data types of the procedure's parameters and the return type (the data type of its result value) must be supported data types. The software must not omit any of the parameters when importing the procedure.
- Data Definition Language (DDL) operations like creating tables, views etc. and transaction control statements such as `COMMIT`/`ROLLBACK` cannot be performed inside the stored procedure.

In addition, when importing the function, you must select the Callable from SQL expression check box on the Import By Name window.

The software will push down a stored procedure call to Oracle if the first five conditions are met and the [Callable from SQL expression](#) check box is selected. If a stored procedure call contains any DDL statements, it ends the current transaction with `COMMIT` or `ROLLBACK`, or it issues any `ALTER SESSION` or `ALTER SYSTEM`

commands. Therefore, you should deselect the *Callable from SQL expression* option on the Import By Name window to ensure that the function call will not be pushed down to database.

When a stored procedure call cannot be pushed down to database, it is submitted to database as a separate SQL statement.

Related Information

[About procedures \[page 787\]](#)

[Calling stored procedures \[page 796\]](#)

25.2.5.3.2 Query and transform output schema

When a stored procedure can provide at least one return value or has any `INOUT` or `OUT` parameters other than `AL_SP_RETCODE` or `AL_SP_ERRMSG`, you can call the procedure by including it as an object in the output schema of a query.

When using a stored procedure in a query output schema, you must map `INOUT` or `OUT` parameters and return values to a column of the output schema. In this case, you cannot map these parameters to variables.

Related Information

[About procedures \[page 787\]](#)

[Calling stored procedures \[page 796\]](#)

25.2.5.4 Without the function wizard

You can enter an existing stored procedure call on the *Mapping* tab or the *Where* tab in a query editor.

When entering a stored procedure call without using the function wizard:

- Match the parameter values you enter to the signature of the stored procedure in Data Services. Values must be specified in the same order that the parameters are defined.
- Use `AL_UNSPECIFIED_PARAM` in place of a missing parameter, such as when you want to use the default value for an `IN` parameter.
- Use the correct naming convention for the stored procedure:
 - `<datastore>.<owner>.<proc_name>`
 - `<datastore>.<owner>.<package>.<proc_name>`.

If the name is case-sensitive in the database (and not all uppercase), enter the name as it appears in the database and use double quotation marks (") around the name to preserve the case.

The two output parameters added by Data Services during import, `AL_SP_RETCODE` and `AL_SP_ERRMSG`, are optional. You do not need to provide arguments for these two parameters if you are not interested in the values stored in them.

Parent topic: [Calling stored procedures \[page 796\]](#)

Related Information

[Stored procedure input parameters \[page 797\]](#)

[Calling stored procedures from scripts \[page 798\]](#)

[Calling stored procedures from queries \[page 799\]](#)

[About procedures \[page 787\]](#)

[Calling stored procedures \[page 796\]](#)

[Checking execution status \[page 802\]](#)

25.2.5.4.1 Including a stored procedure in a query output schema

1. In a data flow diagram, click a query name to open the query editor.
2. In the query editor, right-click the query name in the output schema and select [New Function Call](#).

Alternately, you can right-click an existing stored procedure object in the output schema area of the query editor and select [Modify Function Call](#).

SAP Data Services provides a function wizard to help you include input and output parameters and the return value for a procedure call. The imported stored procedures are listed for each datastore on the first page of the function wizard.

3. Select a function category. Notice that the function names change depending on the category you select.
4. Under Function name, select a stored procedure, then click [Next](#).
5. In the [Define Input Parameter\(s\)](#) window, enter the input parameters for the stored procedure, then click [Next](#).
6. In the [Select Output Parameter](#) window, select the procedure's output parameters that you want to map to the query output.

In the [All output parameters](#) box, select the parameters you want to map, and click the arrow key to move them into the [Selected output parameters](#) box.

You can map more than one INOUT or OUT parameter and the return value as output columns from a single procedure call.

7. Click [Finish](#).

The software adds those parameters to the query's output schema. In this example, the output schema of the Query transform would have two columns, `EMP_NAME` and `EMP_HIREDATE`.

25.2.6 Checking execution status

To allow you to monitor the execution status of a stored procedure, Data Services adds two parameters to the signature of each stored procedure when importing the procedure:

- `AL_SP_RETCODE`, `varchar(256)`. There are three possible values:
 - `ACTA_SP_OK` — The stored procedure call succeeded.
 - `ACTA_SP_DB_CONN_EXCEPTION` — The database connection cannot be created because of a connection error, invalid user, password, or host name.
 - `ACTA_SP_CALL_ERROR` — The connection completes but the call fails in the database. Details for the cause of the error are available in `AL_SP_ERRMSG`.
- `AL_SP_ERRMSG`, `varchar(1024)`

Data Services adds these two parameters to the end of the signature, following any user-defined parameters.

With these parameters, there are two techniques you can use to handle errors if a stored procedure fails during execution:

- Throw an error that stops the job immediately. An error message is logged in the `errorlog.txt` file. To use this method, do not map as output parameters the two parameters Data Services adds to imported stored procedures.
- Save the Oracle error messages into the `AL_SP_ERRMSG` parameter. This method allows the query, script, conditional, or custom function that contains the stored procedure call to continue its execution. To use this method, define parameters for `AL_SP_RETCODE` and `AL_SP_ERRMSG`. Use the execution status stored in `AL_SP_RETCODE` to control your application logic. For example, if you store the value of `AL_SP_ERRMSG` in a variable, you can use the print function to print the variable to the job log. With this method, the error message is not logged in the `errorlog.txt` file.

26 Scripting Language

Use the SAP Data Services scripting language to write scripts, custom functions, and expressions. also known as user-script functions. Further, you can use the scripting language to write expressions such as complex column mapping expressions and WHERE clause conditions.

Custom functions are also known as user-script functions. Further, use the scripting language to write expressions such as complex column mapping expressions and WHERE clause conditions.

The software uses the scripting language internally to save objects that you create into repository tables. Export saved objects to a file with the .atl extension. Later import the file to another repository. You export and import objects for such tasks as migrating to a new environment or backing up a repository before an upgrade.

[Using the scripting language \[page 803\]](#)

Use the smart editor tools to write scripts, custom functions, and expressions.

[Language syntax \[page 804\]](#)

When you use the scripting language, adhere to specific syntax so the objects you are building function correctly. .

[Sample scripts \[page 812\]](#)

To help you create your own scripts, we supply two example scripts.

26.1 Using the scripting language

Use the smart editor tools to write scripts, custom functions, and expressions.

To start writing using the scripting language:

1. Open a data flow or other object In the Designer workspace.
2. Open an applicable object editor. For example, open a script, conditional, or a transform.
3. Inside any of these editors, access the smart editor by clicking the smart editor icon.
4. Start to build your script, custom function, or expression.

Task overview: [Scripting Language \[page 803\]](#)

Related Information

[Language syntax \[page 804\]](#)

[Sample scripts \[page 812\]](#)

[Language syntax \[page 804\]](#)

[Sample scripts \[page 812\]](#)

[Smart Editor and function tools \[page 647\]](#)

26.2 Language syntax

When you use the scripting language, adhere to specific syntax so the objects you are building function correctly. .

Use the syntax from the scripting language in expressions as well as in scripts. With the scripting language, assign values to variables, call functions, and use standard string and mathematical operators. Ensure that you know the proper syntax for statements, columns, table references, strings, variables, and so on.

[Syntax for statements in scripts \[page 805\]](#)

Statements in scripts are detailed steps written in the flow of logic.

[Syntax for column and table references in expressions \[page 805\]](#)

The Data Services scripting language recognizes column and table names without special syntax.

[Strings \[page 806\]](#)

String syntax includes using quotation marks, escape characters, and trailing blanks.

[Variables \[page 807\]](#)

Precede variable names with a dollar sign (\$).

[Variable interpolation \[page 808\]](#)

Embed expressions within constant strings using the correct syntax so that the software correctly evaluates the variables.

[Functions and stored procedures \[page 809\]](#)

Use most built-in functions in scripts, expressions, or custom functions.

[NULL values \[page 809\]](#)

Indicate NULL values using the keyword NULL.

[Validate \[page 811\]](#)

Validate the syntax in your scripts, expressions, and custom functions to ensure that you have created the object correctly.

Parent topic: [Scripting Language \[page 803\]](#)

Related Information

[Using the scripting language \[page 803\]](#)

[Sample scripts \[page 812\]](#)

26.2.1 Syntax for statements in scripts

Statements in scripts are detailed steps written in the flow of logic.

Ensure that you use the following syntax for statements:

- End each statement in a script with a semicolon (;).
- Begin each comment line with a pound sign (#).

Related Information

[Sample scripts \[page 812\]](#)

26.2.2 Syntax for column and table references in expressions

The Data Services scripting language recognizes column and table names without special syntax.

Expressions are a combination of constants, operators, functions, and variables that evaluate to a value of a given data type. Use expressions inside script statements or add them to data flow objects. Because expressions can be used inside data flow objects, they can contain column names.

No special syntax is required for column or table names. For example, you can indicate the `start_date` column as the input to a function as follows:

```
to_char(start_date, 'dd.mm.yyyy')
```

Before you include a column name, ensure that it is a part of the input schema of the query.

If the query contains multiple sources with some of the same column names, ensure that you specify the column in the function with the table name.

❖ Example

Your input schema contains two tables: Status, and Sales. Each table contains a column named `start_date`. You want to include the `start_date` column from the Status table in the function. Write the function as follows:

```
to_char(status.start_date, 'dd.mm.yyyy')
```

Column and table names as part of SQL strings may require special syntax based on the DBMS by which the SQL is evaluated.

❖ Example

Select all rows from the `LAST_NAME` column of the `CUSTOMER` table as follows:

```
sql('oracle_ds', 'SELECT CUSTOMER.LAST_NAME  
FROM CUSTOMER')
```

Because the column name is all upper case in Oracle, it does not require special syntax. However, select all rows from the `start_date` column in the Oracle table `Status` as follows:

```
sql('oracle_ds','SELECT "Status"."start_date"
FROM "Status"')
```

The table name and column are defined in Oracle to have a special case. To preserve this case, enclose the identifiers in double quotation marks.

26.2.3 Strings

String syntax includes using quotation marks, escape characters, and trailing blanks.

- Quotation marks: Choose the type of quotation mark to use based on whether you use identifiers or constants.
- Escape characters: Some characters need escape characters to differentiate the character use in the string.
- Trailing blanks: Remove trailing blanks with built-in functions.

Quotation marks [\[page 806\]](#)

The type of quotation marks to use in strings depends on whether you are using identifiers or constants.

Escape characters [\[page 807\]](#)

Constants that contain single quotes, backslashes, or other special characters use escape characters so that the function knows how to process them.

Trailing blanks [\[page 807\]](#)

SAP Data Services does not strip trailing blanks from strings that you use in scripts or custom functions.

26.2.3.1 Quotation marks

The type of quotation marks to use in strings depends on whether you are using identifiers or constants.

The following table describes the types of quotation marks to use for each string type.

Identifier	Name of an object such as table, column, data flow, or function. Identifiers need quotation marks when they contain special characters that are non alphanumeric.

❖ Example

Use a double quotation for the following string because it contains blanks: `"compute large numbers"`

Constant

A fixed value used in computation. There are two types of constants:

- String constants require single quotation marks. For example: 'Hello World' or '1995.01.23'.
- Numeric constants do not require quotation marks. For example, 2.14.

26.2.3.2 Escape characters

Constants that contain single quotes, backslashes, or other special characters use escape characters so that the function knows how to process them.

When your script uses a syntax character that is not intended as syntax, precede the character with an escape character.

SAP Data Services uses the backslash (\) as the escape character.

Character	Example
Multiple single quotation marks.	'World\'s Books'
Multiple backslashes.	'C:\\temp'

26.2.3.3 Trailing blanks

SAP Data Services does not strip trailing blanks from strings that you use in scripts or custom functions.

To remove the trailing blanks from strings, use the built-in functions `rtrim` or `rtrim_blank`.

Related Information

[rtrim](#)
[rtrim_blanks](#)

26.2.4 Variables

Precede variable names with a dollar sign (\$).

- You define local variables used in a script or expression in the job or work flow that calls the script. Use the [Variables and Parameters](#) window to define the local variables.

- You define global variables used in a script or expression at the job level using the [Variables and Parameters](#) window.
- Use the following statement to ensure that the function passes the return value outside the function:

```
RETURN (<expression>)
```

<expression> contains the definition of the value to be returned.

- Define local variables used in a custom function using the smart editor.
- The smart editor filters the display of existing variables and parameters based on the context from which you open the smart editor.

Related Information

[Variables and Parameters \[page 701\]](#)

[Creating custom functions from the Tools menu \[page 779\]](#)

26.2.5 Variable interpolation

Embed expressions within constant strings using the correct syntax so that the software correctly evaluates the variables.

When you embed expressions within constant strings, the software evaluates the variables and substitutes the value into the string. The software does not need the concatenation operator (||) to make the substitution.

The following expression uses the concatenation operator:

```
$st_date=sql('warehouse_ds','SELECT max(start_timestamp)
FROM dw_process_log
WHERE table_name='\'||$Table_Name||'\')
print('The value of the start date
is: '||sysdate()+5);
```

Simplify the expression as follows excluding the concatenation operator:

```
$st_date=sql('warehouse_ds','SELECT max(start_timestamp)
FROM dw_process_log
WHERE table_name={$Table_Name}')
print('The value of the start date
is:[sysdate()+5]');
```

Use curly braces ({}) and square brackets ([]) to enclose the embedded expressions:

- The square brackets ([]) indicate to substitute the value of the expression.
- The curly braces ({}) indicate to add single quotation marks to the value of the expression.

Strings that include curly braces or square brackets cause processing errors. Avoid the errors by preceding the braces or brackets with a backslash (\).

26.2.6 Functions and stored procedures

Use most built-in functions in scripts, expressions, or custom functions.

Keep in mind that a function cannot call itself or call another function that results in a recursive call. For example, function A cannot call function A, nor can function A call function B if function B calls function A.

You can also use functions and stored procedures imported from a DBMS. Use the exact case for names of imported functions and stored procedures.

Related Information

[Functions and Procedures \[page 773\]](#)

26.2.7 NULL values

Indicate NULL values using the keyword NULL.

For example, you can check whether a column (COLX) is null or not:

```
COLX IS NULL  
COLX IS NOT NULL
```

The software does not check for NULL values in data columns. Use the function `nvl` to remove NULL values.

Related Information

[nvl](#)

26.2.7.1 NULL values and empty strings

The software has specific rules for syntax with NULL values and empty strings.

There are the two rules that regulate empty strings:

- Assigning an empty string to a variable of type varchar.
 - The software treats the value of the variable as a zero length string.
 - If you assign an empty string to a variable that is not a varchar, the software issues an error.
 - Assign a NULL value to a variable of any type using the NULL constant.
- When an empty string is a constant:
 - The software treats an empty string that is a constant (' ') as a varchar value of zero length.

- Use a NULL constant for the NULL value.

i Note

Oracle does not distinguish an empty string from a NULL value. When you insert an empty string or a NULL value into a varchar column, Oracle treats both the empty string and NULL value as NULL values. Therefore, the software treats the value as a NULL value.

NULLs and empty strings in conditionals

There are three rules for NULLs and empty strings in conditionals:

- Equals and Not Equal to comparison operators always evaluate to FALSE against a NULL value.
- Use the IS NULL and IS NOT NULL operators to test the presence of NULL values.
- When comparing two variables, always test for NULL.

Equals (=) and Not Equal (<>) evaluate to FALSE against NULL

The FALSE result includes comparing a variable that has a value of NULL against a NULL constant.

The following table shows the comparison results for the variable assignments \$var1 = NULL and \$var2 = NULL:

Condition	Translates to	Returns
If (NULL = NULL)	NULL is equal to NULL	FALSE
If (NULL != NULL)	NULL is not equal to NULL	FALSE
If (NULL = '')	NULL is equal to empty string	FALSE
If (NULL != '')	NULL is not equal to empty string	FALSE
If ('bbb' = NULL)	bbb is equal to NULL	FALSE
If ('bbb' != NULL)	bbb is not equal to NULL	FALSE
If ('bbb' = '')	bbb is equal to empty string	FALSE
If ('bbb' != '')	bbb is not equal to empty string	TRUE
If (\$var1 = NULL)	NULL is equal to NULL	FALSE
If (\$var != NULL)	NULL is not equal to NULL	FALSE
If (\$var1 = '')	NULL is equal to empty string	FALSE
If (\$var != '')	NULL is not equal to empty string	FALSE
If (\$var1 = \$var2)	NULL is equal to NULL	FALSE
If (\$var != \$var2)	NULL is not equal to NULL	FALSE

The following table shows the comparison results for the variable assignments \$var1 = '' and \$var2 = '':

Condition	Translates to	Returns
If (\$var1 = NULL)	Empty string is equal to NULL	FALSE
If (\$var != NULL)	Empty string is not equal to NULL	FALSE
If (\$var1 = '')	Empty string is equal to empty string	TRUE
If (\$var != '')	Empty string is not equal to empty string	FALSE
If (\$var1 = \$var2)	Empty string is equal to Empty string	TRUE
If (\$var != \$var2)	Empty string is not equal to Empty string	FALSE

Use the IS NULL and IS NOT NULL operators

Use the IS NULL and IS NOT NULL operators to test the presence of null values. For example, assuming a variable is assigned: `$var1 = NULL;`

Condition	Translates to	Returns
If ('bbb' IS NULL)	bbb is NULL	FALSE
If ('bbb' IS NOT NULL)	bbb is not NULL	TRUE
If ('' IS NULL)	Empty string is NULL	FALSE
If ('' IS NOT NULL)	Empty string is not NULL	TRUE
If (\$var1 IS NULL)	NULL is NULL	TRUE
If (\$var IS NOT NULL)	NULL is not NULL	FALSE

When comparing two variables, always test for NULL

In this scenario, you are not testing a variable with a value of NULL against a NULL constant as in the first rule. Either test each variable and branch accordingly or test in the conditional as shown in the second row of the following table.

Condition	Recommendation
<code>if (\$var1 = \$var2)</code>	Do not compare without explicitly testing for NULLs. Using this logic is not recommended because any relational comparison to a NULL value returns FALSE.
<code>if (((\$var1 IS NULL) AND (\$var2 IS NULL)) OR (\$var1 = \$var2))</code>	Executes the TRUE branch if both \$var1 and \$var2 are NULL, or if neither are NULL but are equal to each other.

26.2.8 Validate

Validate the syntax in your scripts, expressions, and custom functions to ensure that you have created the object correctly.

When you create an object using the smart editor, use the [Validate](#) icon in the tool bar to check the correctness of your object. The software lists errors in the smart editor under the editor pane text box. Double-click each error to see the location in your script.

If you are not using smart editor, use the [Validate](#) menu in Designer to validate script objects.

- Select the [Validate > Current View](#) to check the syntax of expressions used in the definition of the current object. The software displays errors in the Output dialog box.
- Select the [Validate > All Objects in View](#) to check the syntax of expressions in the current object definition and the definitions of all objects called by the current object. The software displays errors in the Output dialog box.

By default, Data Services lists ten errors before it aborts parsing. By reporting ten errors for each validation, Data Services allows you to shorten the edit-validate-test cycle. You can repair several errors in each iteration of the cycle.

To change the number of reported errors from ten, select [Tools > Options > Job Server > General](#) and set the options as follows:

Section:	Enter "Parser".
Key:	Enter "NumErrors".
Value:	Enter a positive number.

For each error, Data Services provides a description of the problem.

Data Services also provides three menu options for errors. Right-click an error to use these menu options:

- View: The [View](#) option allows you to view the error text in a smaller, separate window for easy reading.
- Go To Error: The [Go to Error](#) option places an error icon near the line that caused the error. You can also double-click the error to place this icon. This option not available for all errors, however.
- Copy: The [Copy](#) option copies the error text on to the clipboard so you can place it elsewhere.

When you execute a job, you may still encounter errors in an expression that are returned from the operating system or the DBMS. When executing a job, Data Services sends as many operations as possible to the DBMS to evaluate. The DBMS may evaluate part of the expression and return a value that contributes to evaluating the rest of the expression.

26.3 Sample scripts

To help you create your own scripts, we supply two example scripts.

The examples show how to format a script for the Square and Repeat String functions.

Parent topic: [Scripting Language \[page 803\]](#)

Related Information

[Using the scripting language \[page 803\]](#)

[Language syntax \[page 804\]](#)

26.3.1 Square function

The square function returns the square value of an input integer.

To define the script, use the correct syntax in the square function using the following values:

Description	Value	Arguments
Function name	Square	
Return value	Return	int
Input value	input_int	int, input

The text of the function script is as follows:

```
RETURN($input_int * $input_int);
```

26.3.2 RepeatString function

Use a RepeatString function to repeat a given string x number of times in the output.

This function accepts a one-character string and an integer indicating the number of times to repeat the input character. It outputs the created string or, if the input character or repeat number is NULL, it outputs a NULL value. It uses a variable to track the string components as it generates the string.

To define this script, supply the following values to Data Services:

Description	Value	Arguments
Function name	RepeatString	
Return value	Return	VARCHAR (255),allow NULL
Input string	InputCharacter	VARCHAR (1), input, allow NULL
Input integer	RepeatNumber	INT, input, allow NULL
Internal variable	Partial	VARCHAR (255), allow NULL

The following code shows the function with the correct syntax:

```
IF ($InputCharacter = NULL)
BEGIN
    $Partial = NULL;
    RETURN NULL;
END
IF ($RepeatNumber = NULL)
    RETURN NULL;
IF ($RepeatNumber > 255)
    $RepeatNumber = 255;
IF ($RepeatNumber < 0)
    $RepeatNumber = 0;
BEGIN
    WHILE ($RepeatNumber != 0)
    BEGIN
        $Partial = ($Partial || $InputCharacter);
        $RepeatNumber = ($RepeatNumber - 1);
    END
END
```

```
END  
RETURN $Partial;  
END
```

27 Design and Debug

This section covers the following Designer features that you can use to design and debug jobs:

- Use the View Where Used feature to determine the impact of editing a metadata object (for example, at table). See which data flows use the same object.
- Use the View Data feature to view sample source, transform, and target data in a data flow after a job executes.
- Use the Design-Time Data Viewer feature to view and analyze the input and output for a data set in real time as you design a transform.
- Use the Interactive Debugger to set breakpoints and filters between transforms within a data flow and view job data row-by-row during a job execution.
- Use the Difference Viewer to compare the metadata for similar objects and their properties.
- Use the auditing data flow feature to verify that correct data is processed by a source, transform, or target object.

Related Information

[Using View Where Used \[page 815\]](#)

[Using View Data \[page 819\]](#)

[Using the Design-Time Data Viewer \[page 831\]](#)

[Using the interactive debugger \[page 832\]](#)

[Comparing Objects \[page 846\]](#)

[Using Auditing \[page 335\]](#)

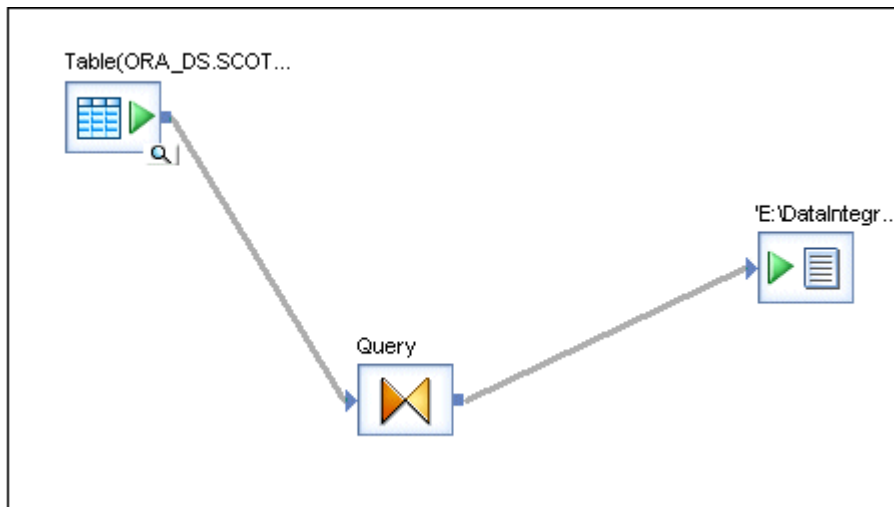
27.1 Using View Where Used

When you save a job, work flow, or data flow the software saves a list of related objects in your repository.

The software preserves parent-child relationship data. You can view relationships using the [Tools](#) menu ( [Tools](#) > [View Where Used](#) ). You can also access the information by clicking the [View Where Used](#) icon in the tool bar.

For example, when the following parent data flow is saved, the software also saves connections between it and its three children:

- a table source
- a query transform
- a file target



You can use this parent/child relationship data to determine what impact a potential change will have on the other data flows that use the same object.

For example, while maintaining a data flow, you may need to delete a source table definition and re-import the table (or edit the table schema). Before doing this, find all the data flows that are using the same table and update them as needed.

The Designer counts an object's usage as the number of times it is used for a unique purpose.

For example, in a data flow named DF1, if table DEPT is used as a source twice and a target once, the number in the [Usage](#) column in the object library is 2. This occurrence should be rare. For example, a table is not often joined to itself in a job design

Related Information

[Limitations \[page 819\]](#)

27.1.1 Accessing View Where Used from the object library

The Designer object library contains a [Usage](#) column that shows the number of parent-child relationship for an object.

View parent-child relationships for the following object types:

- Jobs
- Work flows
- Data flows
- Transforms (Data Quality, Text Data Processing, and ABAP transforms in Platform folder)
- Datastores
- Formats

When the usage value is greater than zero, there is relationship information available for you to view.

1. Right-click the object name in the object library and select [View Where Used](#) from the drop menu.

The [Output](#) dialog opens in the lower portion of the workspace.

- The [Information](#) tab contains rows for each parent of the object you selected.
- The columns display the object type and name.
- The [As](#) column tells you how the selected object is used by the parent.

2. Double-click a parent object to view it in the workspace.

The workspace shows the object diagram with the child object highlighted.

Once a parent is open in the workspace, you can double-click a row in the output window again to see usage information for a different object.

Object	Work space shows
Parent	Diagram for the parent object.
Child	If it is a child object in the same parent, the highlighted object changes to the child in the diagram.

→ Tip

This is a useful feature. A child object in the [Output](#) dialog might not match the name used in its parent. You can customize workspace object names for sources and targets. The software saves both the name used in each parent and the name used in the object library. The Information tab on the [Output](#) window displays the name used in the object library. The names of objects used in parents can only be seen by opening the parent in the workspace.

3. Select an object in a different row of the [Output](#) dialog. The workspace changes to display the diagram of the selected object.

Related Information

[About the Output dialog \[page 818\]](#)

[Limitations \[page 819\]](#)

[Accessing View Where Used for variables and parameters \[page 706\]](#)

27.1.1.1 About the Output dialog

The information in the [Output](#) dialog for [View Where Used](#) can be very helpful when you decide to make changes to a specific object.

Open the [Output](#) dialog by right-clicking an object name that has more than zero (0) in the [Usage](#) column in the Designer object library.

The [Information](#) tab in the [Output](#) dialog displays the usage information. It contains a row for each parent of the object you selected. The dialog displays the type and name of the selected object in the first column's heading. The [As](#) column provides additional context about how the selected object is used by the parent, when applicable. For example, XML files and messages, tables, flat files, and so on, can have a value of "Source" or "Target" in the [As](#) column meaning the object is used as a source or target.

The following tables show other potential values in the [As](#) column.

Flat files and tables only

As	Description
Lookup()	Lookup table or file is used in a <code>lookup</code> function
Lookup_ext()	Lookup table or file is used in a <code>lookup_ext</code> function
Lookup_seq()	Lookup table or file is used in a <code>lookup_seq</code> function

Tables only

As	Description
Comparison	Table used in the Table Comparison transform
Key Generation	Table used in the Key Generation transform

27.1.2 Accessing View Where Used from the workspace

From an open diagram of an object in the workspace (such as a data flow), you can view where a parent or child object is used.

Output window lists	How to access View Where Used
Jobs that use a specific data flow	Open a data flow in the workspace. Select  View  Where Used  from the menu bar or select the View Where Used button in the toolbar.

Output window lists

List of parent objects that use a selected object in a data flow.

How to access View Where Used

Open a data flow in the workspace and select the applicable data flow object. Select **View > Where Used** from the menu bar or select the **View Where Used** button in the tool-bar.

Related Information

[Accessing View Where Used for variables and parameters \[page 706\]](#)

27.1.3 Limitations

Important things to know when you use *View Where Used*.

- The software does not support *View Where Used* in central repositories.
- The *Information* tab of the *Output* window only shows parent and child pairs.

❖ Example

For example, for a table, a data flow is the parent. If the table is also used by a grandparent (a work flow for example), the workflow is not listed in the *Output* window display for a table. To see the relationship between a data flow and a work flow, open the work flow in the workspace, then right-click a data flow and select the *View Where Used* option

- The software does not save parent/child relationships between functions. For example:

❖ Example

If function A calls function B, and function A is not in any data flows or scripts, the number in the *Usage* column in the object library is zero for both functions. The fact that function B is used once in function A is not counted.

❖ Example

If function A is saved in one data flow, the number in the *Usage* column in the object library is 1 for both functions A and B.

27.2 Using View Data

View Data provides a way to scan and capture a sample of the data produced by each step in a job, even when the job does not execute successfully. View imported source data, changed data from transformations, and

ending data at your targets. At any point after you import a data source, you can check on the status of that data—before and after processing your data flows.

Use View Data to check the data while designing and testing jobs to ensure that your design returns the results you expect. Using one or more View Data panes, you can view and compare sample data from different steps. View Data information is displayed in embedded panels for easy navigation between your flows and the data.

Use View Data to look at:

- Sources and targets
View Data allows you to see data before you execute a job. Armed with data details, you can create higher quality job designs. You can scan and analyze imported table and file data from the object library as well as see the data for those same objects within existing jobs. After you execute the job, you can refer back to the source data again.
- Transforms
- Lines in a diagram

i Note

- View Data displays blob data as `<blob>`.
- View Data is not supported for SAP IDocs. For SAP and PeopleSoft, the Table Profile tab and Column Profile tab options are not supported for hierarchies.

Related Information

[Viewing data passed by transforms \[page 844\]](#)

[Using the interactive debugger \[page 832\]](#)

27.2.1 Accessing View Data

27.2.1.1 Viewing data for sources and targets

You can view data for sources and targets from two different locations:

1. [View Data button](#)

View Data buttons appear on source and target objects when you drag them into the workspace. Click the View data button (magnifying glass icon) to open a View Data pane for that source or target object.

2. [Object library](#)

View Data in potential source or target objects from the Datastores or Formats tabs.

Open a View Data pane from the object library in one of the following ways:

- Right-click a table object and select [View Data](#).
- Right-click a table and select [Open](#) or [Properties](#).

The Table Metadata, XML Format Editor, or Properties window opens. From any of these windows, you can select the View Data tab.

To view data for a file, the file must physically exist and be available from your computer's operating system. To view data for a table, the table must be from a supported database.

Related Information

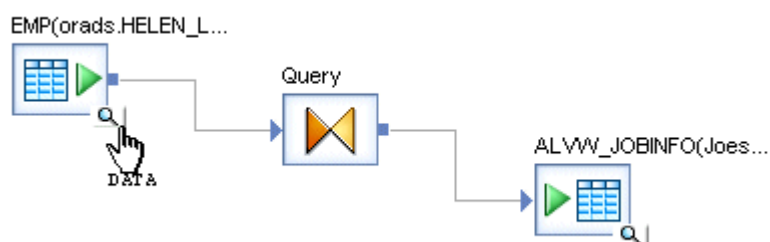
[Viewing data in the workspace \[page 821\]](#)

27.2.2 Viewing data in the workspace

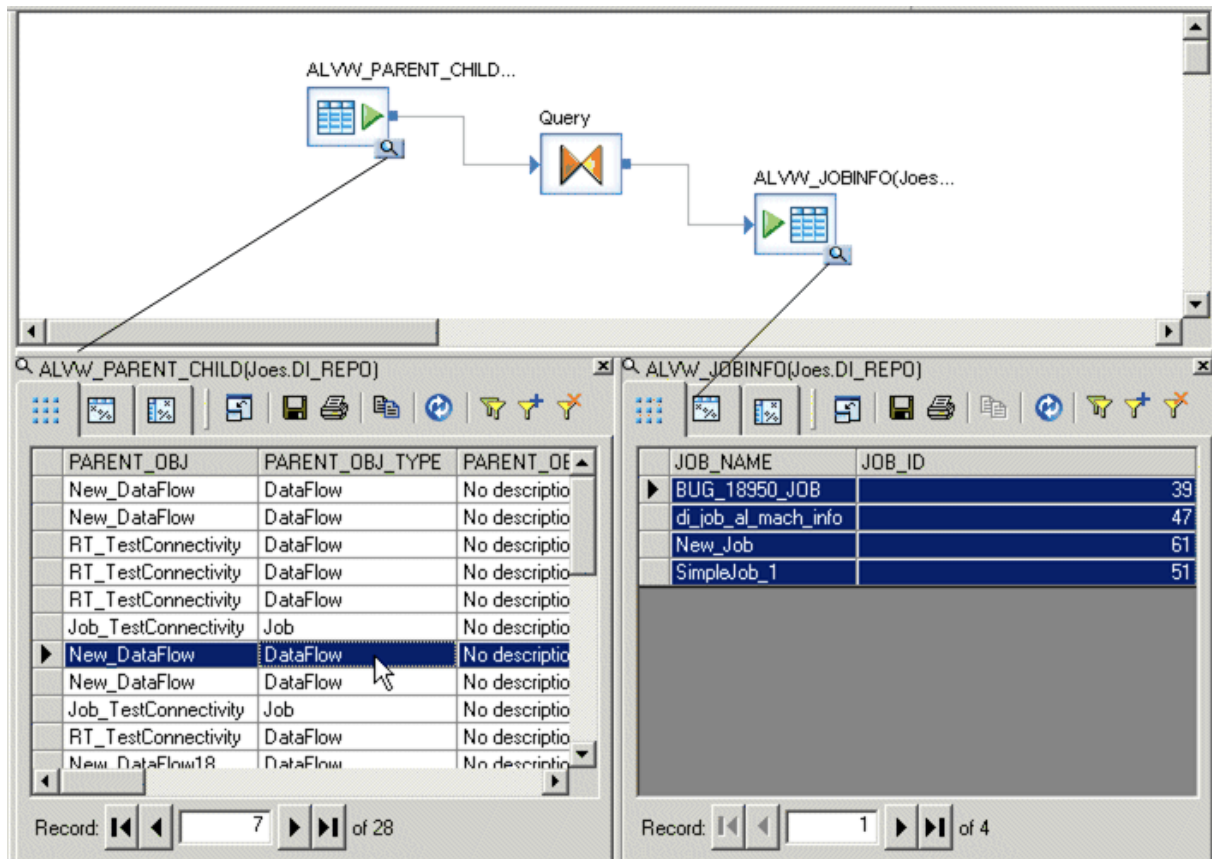
View Data can be accessed from the workspace when magnifying glass buttons appear over qualified objects in a data flow. This means:

For sources and targets, files must physically exist and be available from the Designer, and tables must be from a supported database.

To open a View Data pane in the Designer workspace, click the magnifying glass button on a data flow object.



A large View Data pane appears beneath the current workspace area. Click the magnifying glass button for another object and a second pane appears below the workspace area. (Note that the first pane area shrinks to accommodate the presence of the second pane).



You can open two View Data panes for simultaneous viewing. When both panes are filled and you click another View Data button, a small menu appears containing window placement icons. The black area in each icon indicates the pane you want to replace with a new set of data. Click a menu option and the data from the latest selected object replaces the data in the corresponding pane.

The description or path for the selected View Data button displays at the top of the pane.

- For sources and targets, the description is the full object name:
 - <ObjectName> (<Datastore.Owner>) for tables
 - <FileName> (<File Format Name>) for files
 - For View Data buttons on a line, the path consists of the object name on the left, an arrow, and the object name to the right.
- For example, if you select a View Data button on the line between the query named Query and the target named ALVW_JOBINFO(joes.DI_REPO), the path would indicate:

```
Query -> ALVW_JOBINFO(Joes.DI_REPO)
```

You can also find the View Data pane that is associated with an object or line by:

- Rolling your cursor over a View Data button on an object or line. The Designer highlights the View Data pane for the object.
- Looking for grey View Data buttons on objects and lines. The Designer displays View Data buttons on open objects with grey rather than white backgrounds.

Related Information

[Viewing data passed by transforms \[page 844\]](#)

27.2.3 View Data Properties

You can access View Data properties from tool bar buttons or the right-click menu.

View Data displays your data in the rows and columns of a data grid. The number of rows displayed is determined by a combination of several conditions:

- Sample size — The number of rows sampled in memory. Default sample size is 1000 rows for imported source and target objects. Maximum sample size is 5000 rows. Set sample size for sources and targets from ► [Tools](#) ► [Options](#) ► [Designer](#) ► [General](#) ► [View Data sampling size](#) . When using the interactive debugger, the software uses the Data sample rate option instead of sample size.
- Filtering
- Sorting

If your original data set is smaller or if you use filters, the number of returned rows could be less than the default.

You can see which conditions have been applied in the [navigation bar](#).

Related Information

[Filtering \[page 823\]](#)

[Sorting \[page 825\]](#)

[Starting and stopping the interactive debugger \[page 836\]](#)

27.2.3.1 Filtering

You can focus on different sets of rows in a local or new data sample by placing fetch conditions on columns.

27.2.3.1.1 Viewing and adding filters

1. In the View Data tool bar, click the Filters button, or right-click the grid and select [Filters](#). 

The Filters window opens.

2. Create filters.

The Filters window has three columns:

- a. Column—Select a name from the first column. Select *{remove filter}* to delete the filter.
- b. Operator—Select an operator from the second column.
- c. Value—Enter a value in the third column that uses one of the following data type formats

Data Type	Format
Integer, double, real	standard
date	yyyy.mm.dd
time	hh24:mm:ss
datetime	yyyy.mm.dd hh24:mm:ss
varchar	'abc'

3. In the *Concatenate all filters using* list box, select an operator (*AND*, *OR*) for the engine to use in concatenating filters.

Each row in this window is considered a filter.

4. To see how the filter affects the current set of returned rows, click *Apply*.
5. To save filters and close the Filters window, click *OK*.

Your filters are saved for the current object and the local sample updates to show the data filtered as specified in the Filters dialog. To use filters with a new sample, see *Using Refresh*.

Related Information

[Using Refresh \[page 825\]](#)

27.2.3.1.2 Adding a filter for a selected cell

1. Select a cell from the sample data grid.

2. In the View Data tool bar, click the Add Filter button, or right-click the cell and select *Add Filter*. 

The Add Filter option adds the new filter condition, <column> = <cell value>, then opens the Filters window so you can view or edit the new filter.

3. When you are finished, click *OK*. 

To remove filters from an object, go to the View Data tool bar and click the Remove Filters button, or right-click the grid and select *Remove Filters*. All filters are removed for the current object.

27.2.3.2 Sorting

You can click one or more column headings in the data grid to sort your data. An arrow appears on the heading to indicate sort order: ascending (up arrow) or descending (down arrow).

To change sort order, click the column heading again. The priority of a sort is from left to right on the grid.



To remove sorting for an object, from the tool bar click the Remove Sort button, or right-click the grid and select [Remove Sort](#).

Related Information

[Using Refresh \[page 825\]](#)

27.2.3.3 Using Refresh



To fetch another data sample from the database using new filter and sort settings, use the [Refresh](#) command. After you edit filtering and sorting, in the tool bar click the Refresh button in the tool bar, or right-click the data grid and select [Refresh](#).



To stop a refresh operation, click the Stop button. While the software is refreshing the data, all View Data controls except the [Stop](#) button are disabled.

27.2.3.4 Using Show/Hide Columns

You can limit the number of columns displayed in View Data by using the [Show/Hide Columns](#) option from:

- The tool bar.
- The right-click menu.
- The arrow shortcut menu, located to the right of the [Show/Hide Columns](#) tool bar button. This option is only available if the total number of columns in the table is ten or fewer. Select a column to display it.

You can also "quick hide" a column by right-clicking the column heading and selecting [Hide](#) from the menu.

27.2.3.4.1 Showing or hiding columns

-  Click the Show/Hide columns tool bar button, or right-click the data grid and select [Show/Hide Columns](#).
The Column Settings window opens.
- Select the columns that you want to display or click one of the following buttons: [Show](#), [Show All](#), [Hide](#), or [Hide All](#).
- Click [OK](#).

27.2.3.5 Opening a new window



To see more of the data sample that you are viewing in a View Data pane, open a full-sized View Data window. From any View Data pane, click the Open Window tool bar button to activate a separate, full-sized View Data window. Alternatively, you can right-click and select [Open in new window](#) from the menu.

27.2.4 View Data tool bar options

The following options are available on View Data panes.

Icon	Option	Description
	Open in new window	Opens the View Data pane in a larger window.
	Save As	Saves the data in the View Data pane.
	Print	Prints View Data pane data.
	Copy Cell	Copies View Data pane cell data.
	Refresh data	Fetches another data sample from existing data in the View Data pane using new filter and sort settings.
	Open Filters window	Opens the Filters window.
	Add a Filter	Adds a filter to a selected cell.
	Remove Filter	Removes all filters in the View Data pane.

Icon	Option	Description
	Remove Sort	Removes sort settings for the object you select.
	Show/Hide navigation	Shows or hides the navigation bar which appears below the data table.
	Show/Hide columns	

Related Information

[Opening a new window \[page 826\]](#)

[Using Refresh \[page 825\]](#)

[Filtering \[page 823\]](#)

[Sorting \[page 825\]](#)

[Adding a filter for a selected cell \[page 824\]](#)

[Using Show/Hide Columns \[page 825\]](#)

27.2.5 View Data tabs

The View Data panel for objects contains three tabs:

- Data tab
- Profile tab
- Column Profile tab

Use tab options to give you a complete profile of a source or target object. The Data tab is always available. The Profile and Relationship tabs are supported with the Data Profiler. Without the Data Profiler, the Profile and Column Profile tabs are supported for some sources and targets.

Related Information

[Viewing the profiler results \[page 320\]](#)

27.2.5.1 Data tab

The Data tab allows you to use the properties of [View Data](#). It also indicates nested schemas such as those used in XML files and messages. When a column references nested schemas, that column is shaded yellow and a small table icon appears in the column heading.

Related Information

[View Data Properties \[page 823\]](#)

27.2.5.1.1 Viewing a nested schema

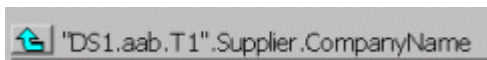
1. Double-click a cell.

The data grid updates to show the data in the selected cell or nested table. 

In the Schema area, the selected cell value is marked by a special icon. Also, tables and columns in the selected path are displayed in blue, while nested schema references are displayed in grey.

In the Data area, data is shown for columns. Nested schema references are shown in angle brackets; for example, `<CompanyName>`.

2. Continue to use the data grid side of the panel to navigate. For example:
 - Select a lower-level nested column and double-click a cell to update the data grid.
 - Click the at the top of the data grid to move up in the hierarchy.



- See the entire path to the selected column or table displayed to the right of the Drill Up button. Use the path and the data grid to navigate through nested schemas.

27.2.5.2 Profile tab

If you use the Data Profiler, the Profile tab displays the profile attributes that you selected on the [Submit Column Profile Request](#) option.

The Profile tab allows you to calculate statistical information for any set of columns you choose. This optional feature is not available for columns with nested schemas or for the LONG data type.

Related Information

[Executing a profiler task \[page 314\]](#)

27.2.5.2.1 Using the Profile tab without the Data Profiler

1. Select one or more columns.

Select only the column names you need for this profiling operation because Update calculations impact performance.

You can also right-click to use the Select All and Deselect All menu options.

2. Click [Update](#).
3. The statistics appear in the Profile grid.

The grid contains six columns:

Column	Description
Column	Names of columns in the current table. Select names from this column, then click Update to populate the profile grid.
Distinct Values	The total number of distinct values in this column.
NULLs	The total number of NULL values in this column.
Min	Of all values, the minimum value in this column.
Max	Of all values, the maximum value in this column.
Last Updated	The time that this statistic was calculated.

Sort values in this grid by clicking the column headings. Note that Min and Max columns are not sortable.

In addition to updating statistics, you can click the [Records](#) button on the Profile tab to count the total number of physical records in the object you are profiling.

The software saves previously calculated values in the repository and displays them until the next update.

27.2.5.3 Column Profile tab

The Column Profile tab allows you to calculate statistical information for a single column. If you use the Data Profiler, the Relationship tab displays instead of the Column Profile.

i Note

This optional feature is not available for columns with nested schemas or the LONG data type.

Related Information

[Viewing the relationship profile data generated by the Data Profiler \[page 324\]](#)

27.2.5.3.1 Calculating value usage statistics for a column

1. Enter a number in the *Top* box.

This number is used to find the most frequently used values in the column. The default is 10, which means that the software returns the top 10 most frequently used values.

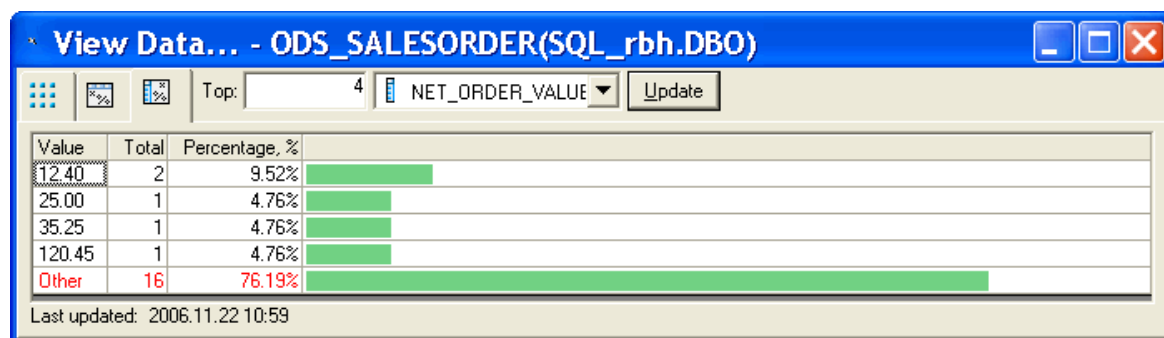
2. Select a column name in the list box.
3. Click *Update*.

The Column Profile grid displays statistics for the specified column. The grid contains three columns:

Column	Description
Value	A top (most frequently used) value found in your specified column, or <i>Other</i> (remaining values that are not used as frequently).
Total	The total number of rows in the specified column that contain this value.
Percentage	The percentage of rows in the specified column that have this value compared to the total number of values in the column.

The software returns a number of values up to the number specified in the *Top* box plus an additional value called *Other*.

So if you enter 4 in the Top box, you will get up to five returned values (the top-four-used values in the specified column, plus the Other category). Results are saved in the repository and displayed until you perform a new update.



For example, statistical results in the preceding table indicate that of the four most frequently used values in the NET_ORDR_VALUE column, 9.52% use the value 12.40, 4.76% use the value 25.00, and so on. You can also see that the four most frequently used values (the "top four") are used in approximately 24% of all cases because approximately 76% is shown in the Other category. For this example, the total number of rows counted during the calculation for each top value is 21.

27.3 Using the Design-Time Data Viewer

The Design-Time Data Viewer lets you view and analyze the input and output for a data set in real time as you design a transform. The data flow does not need to be complete or valid, although it must use a valid, available source that contains data.

Use the Design-Time Data Viewer to view the data while designing a transform to ensure that your design returns the results that you expect. The Design-Time Data Viewer displays as input and output panes in the transform editor so that you can compare the data before and after the transform acts on it.

To use the Design-Time Data Viewer for the Global Address Cleanse, Geocoder, and USA Regulatory Address Cleanse transforms, you must have access to the Data Quality reference data (directories).

Related Information

[Viewing Design-Time Data \[page 831\]](#)

[Configuring the Design-Time Data Viewer \[page 832\]](#)

27.3.1 Viewing Design-Time Data

1. To view the input and output data for transforms in the transform editor, select **Debug > View Design-Time Data**.

Input and output data panes open in the transform editor. Each pane may contain several tabbed views depending on how many inputs and outputs the selected transform has. For example, although most transforms have only one output, the Validation transform has three outputs.

2. To view the data for a different input or output, click the appropriate tab in the View Data pane.
3. To view the input and output data automatically after you modify a transform, select **Debug > View Automatically**.
4. To filter the number of data rows displayed in the panes, select **Debug > Filter Input Dataset**. Setting the filter may help increase performance while you are designing the transform and have the Design-Time Data Viewer feature set to update automatically.

By default, the filter displays the first 50 rows. You can configure the number of rows that are displayed in the *Options* window.

5. To close the Design-Time Data Viewer panes, click the x in the upper-right corner of the pane.

Related Information

[Configuring the Design-Time Data Viewer \[page 832\]](#)

27.3.2 Configuring the Design-Time Data Viewer

You can configure the number of data rows that are displayed in the Design-Time Data Viewer panes as well as the time that is allowed for updates before it times out.

1. To configure the Design-Time Data Viewer options, select . A window opens and displays the available options.
2. Edit the options as necessary.

Option	Description
Number of rows to read from any source when filtered	Specifies the number of rows that are read from the source if the Filter Input Dataset menu item is selected. By default, the filter reads the first 50 rows of data from the source.
Time-out interval for automatic mode	Specifies the amount of time, in seconds, allowed to update the data if the View Automatically menu item is selected before timing out and returning an error.
Time-out interval for manual mode	Specifies the amount of time, in seconds, allowed to update the data if the View Automatically menu item is deselected before timing out and returning an error.

3. Click [OK](#) to close the window.

27.3.3 Specifying variables for expressions

You can specify variables for use in transformation expressions. The value of the variable is used in design-time data calculations. You can also set values for global variables that are used only for design-time calculations.

1. To configure the Design-Time Data Viewer options, select . A window opens and displays the available options.
2. In the Variables area, enter the name, data type, and value for the variable.
3. To import a global variable, click the [Import](#) button. All global variables from each job in the repository populate the table.
4. Enter the value for each imported global value.
Variable values set in the Design-Time Data Viewer options are only used for design-time data calculations.
5. To remove a variable from the table, select it and click the [Delete](#) button.
6. Click [OK](#) to close the window.

27.4 Using the interactive debugger

The Designer includes an interactive debugger that allows you to examine and modify data row-by-row (during a debug mode job execution) by placing filters and breakpoints on lines in a data flow diagram. The interactive debugger provides powerful options to debug a job.

i Note

A repository upgrade is required to use this feature.

27.4.1 Before starting the interactive debugger

Like executing a job, you can start the interactive debugger from the [Debug](#) menu when a job is active in the workspace. Select [Start debug](#), set properties for the execution, then click [OK](#). The debug mode begins. The Debug mode provides the interactive debugger's windows, menus, and tool bar buttons that you can use to control the pace of the job and view data by pausing the job execution using filters and breakpoints.

While in debug mode, all other Designer features are set to read-only. To exit the debug mode and return other Designer features to read/write, click the [Stop debug](#) button on the interactive debugger toolbar.

All interactive debugger commands are listed in the Designer's [Debug](#) menu. The Designer enables the appropriate commands as you progress through an interactive debugging session.

Before you start a debugging session, however, you might want to set the following:

- Filters and breakpoints
- Interactive debugger port between the Designer and an engine.

27.4.1.1 Setting filters and breakpoints

You can set any combination of filters and breakpoints in a data flow before you start the interactive debugger. The debugger uses the filters and pauses at the breakpoints you set.

If you do not set predefined filters or breakpoints:

- The Designer will optimize the debug job execution. This often means that the first transform in each data flow of a job is pushed down to the source database. Consequently, you cannot view the data in a job between its source and the first transform unless you set a predefined breakpoint on that line.
- You can pause a job manually by using a debug option called [Pause Debug](#) (the job pauses before it encounters the next transform).

Related Information

[Push-down optimizer \[page 845\]](#)

27.4.1.1.1 Setting a filter or breakpoint

1. In the workspace, open the job that you want to debug.

- Open one of its data flows.
- Right-click the line that you want to examine and select [Set Filter/Breakpoint](#).

A line is a line between two objects in a workspace diagram.

The Breakpoint window opens. Its title bar displays the objects to which the line connects.

- Set and enable a filter or a breakpoint using the options in this window.

A debug filter functions as a simple Query transform with a WHERE clause. Use a filter to reduce a data set in a debug job execution. Note that complex expressions are not supported in a debug filter.

Place a debug filter on a line between a source and a transform or two transforms. If you set a filter and a breakpoint on the same line, The software applies the filter first. The breakpoint can only see the filtered rows.

Like a filter, you can set a breakpoint between a source and transform or two transforms. A breakpoint is the location where a debug job execution pauses and returns control to you.

Choose to use a breakpoint with or without conditions.

- If you use a breakpoint without a condition, the job execution pauses for the first row passed to a breakpoint.
- If you use a breakpoint with a condition, the job execution pauses for the first row passed to the breakpoint that meets the condition.

A breakpoint condition applies to the after image for UPDATE, NORMAL and INSERT row types and to the before image for a DELETE row type.

Instead of selecting a conditional or unconditional breakpoint, you can also use the Break after 'n' row(s) option. In this case, the execution pauses when the number of rows you specify pass through the breakpoint.

- Click [OK](#).

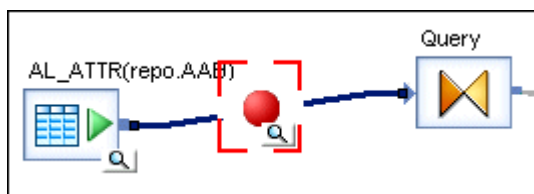
The Breakpoint enabled icon appears on the selected line.

The software provides the following filter and breakpoint conditions:

Icon	Description
	Breakpoint disabled
	Breakpoint enabled
	Filter disabled
	Filter enabled
	Filter and breakpoint disabled
	Filter and breakpoint enabled

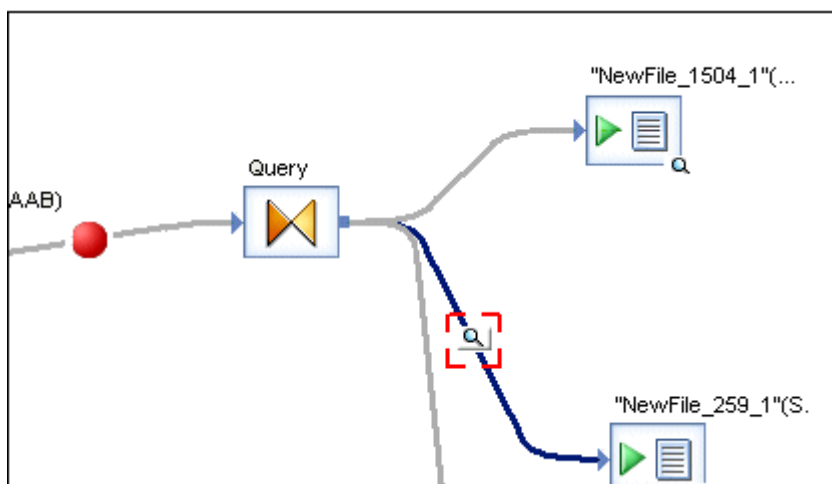
Icon	Description
	Filter enabled and breakpoint disabled
	Filter disabled and breakpoint enabled

In addition to the filter and breakpoint icons that can appear on a line, the debugger highlights a line when it pauses there. A red locator box also indicates your current location in the data flow. For example, when you start the interactive debugger, the job pauses at your breakpoint. The locator box appears over the breakpoint icon as shown in the following diagram:



A View Data button also appears over the breakpoint. You can use this button to open and close the View Data panes.

As the debugger steps through your job's data flow logic, it highlights subsequent lines and displays the locator box at your current position.



Related Information

[Panels \[page 837\]](#)

27.4.1.2 Changing the interactive debugger port

The Designer uses a port to an engine to start and stop the interactive debugger. The interactive debugger port is set to 5001 by default.

27.4.1.2.1 Changing the interactive debugger port setting

1. Select **Tools > Options > Designer > Environment**.
2. Enter a value in the *Interactive Debugger* box.
3. Click *OK*.

27.4.2 Starting and stopping the interactive debugger

A job must be active in the workspace before you can start the interactive debugger. You can select a job from the object library or from the project area to activate it in the workspace. Once a job is active, the Designer enables the *Start Debug* option on the *Debug* menu and tool bar.

27.4.2.1 Starting the interactive debugger

1. In the project area, right-click a job and select *Start debug*.

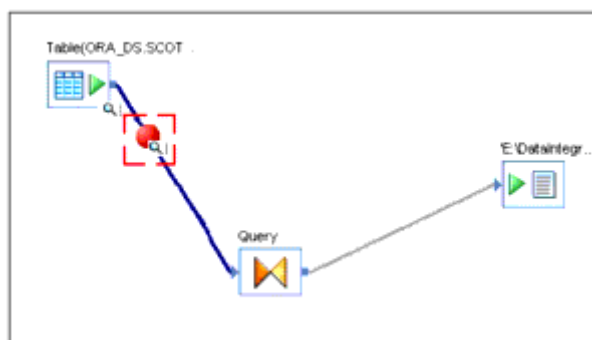
Alternatively, in the project area you can click a job and then:

- Press **Ctrl+F8**.
- From the *Debug* menu, click *Start debug*.
- Click the *Start debug* button on the tool bar.

You will also find more information about the Trace and Global Variable options. The Debug Properties window opens. The Debug Properties window includes three parameters similar to the Execution Properties window (used when you just want to run a job).

The options unique to the Debug Properties window are:

- *Data sample rate*—The number of rows cached for each line when a job executes using the interactive debugger. For example, in the following data flow diagram, if the source table has 1000 rows and you set the *Data sample rate* to 500, then the Designer displays up to 500 of the last rows that pass through a selected line. The debugger displays the last row processed when it reaches a breakpoint.



- [Exit the debugger when the job is finished](#)—Click to stop the debugger and return to normal mode after the job executes. Defaults to cleared.
- 2. Enter the debug properties that you want to use or use the defaults.
- 3. Click [OK](#).

The job you selected from the project area starts to run in debug mode. The Designer:

- Displays the interactive debugger windows.
- Adds Debugging Job [<JobName>](#) to its title bar.
- Enables the appropriate [Debug](#)The Debug Properties window opens. The Debug menu and tool bar options.



- Displays the debug icon in the status bar.
- Sets the user interface to read-only.

i Note

You cannot perform any operations that affect your repository (such as dropping objects into a data flow) when you execute a job in debug mode.

When the debugger encounters a breakpoint, it pauses the job execution. You now have control of the job execution. The interactive debugger windows display information about the job execution up to this point. They also update as you manually step through the job or allow the debugger to continue the execution.

Related Information

[Execution Options \[page 756\]](#)

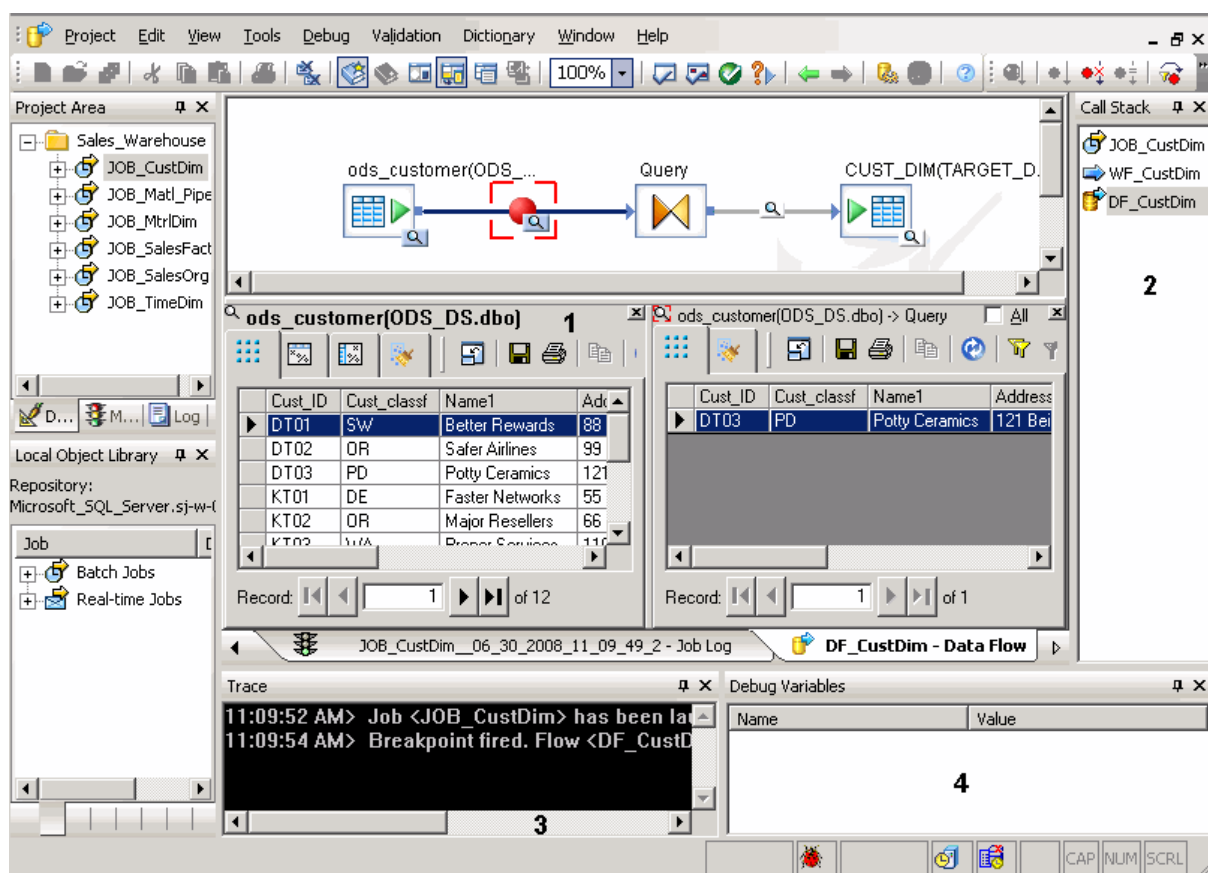
27.4.2.2 Stopping a job in debug mode and exiting the interactive debugger



Click the [Stop Debug](#) button on the tool bar, press [Shift+F8](#), or from the [Debug](#) menu, click [Stop debug](#).

27.4.3 Panes

When you start a job in the interactive debugger, the Designer displays three additional panes as well as the View Data panes beneath the work space. The following diagram shows the default locations for these panes.



1. View Data panes
2. Call Stack pane
3. Trace pane
4. Debug Variable pane

Each pane is docked in the Designer's window. To move a debugger pane, double-click its control bar to release it, then click and drag its title bar to re-dock it.

The Designer saves the layout you create when you stop the interactive debugger. Your layout is preserved for your next Designer session.

You can resize or hide a debugger pane using its control buttons. To show or hide a debugger pane manually, use the [Debug](#) menu or the tool bar.

Related Information

[Debug menu options and tool bar \[page 842\]](#)

27.4.3.1 Call stack window

The Call Stack window lists the objects in the path encountered so far (before either the job completes, encounters a breakpoint, or you pause it).

For example, for the job JOB_CustDim that includes a work flow and data flow, the Call Stack window might display:

JOB_CustDim

WF_CustDim

DF_CustDim

You can double-click an object in the Call Stack window to open it in the workspace. Similarly, if you click an object in a diagram, the Call Stack window highlights the object.

27.4.3.2 Trace window

The Trace window displays the debugger's output status and errors. For example:

```
11:22:06 AM> Job <GO> has been launched in debug mode.  
11:22:07 AM> Breakpoint fired. Flow <aaa>: from <Query> to  
<"NewFile_773" (Simple)>
```

When the job completes, this window displays the following:

Job <JobName> finished. Stop debugger.

When the job completes the debugger gives you a final opportunity to examine data. When you must exit the debugger, select the [Stop Debug](#) button on the tool bar, press [shift+F8](#), or select [DebugStop Debug](#).

27.4.3.3 Debug Variables window

The Debug Variables window displays global variables in use by the job at each breakpoint.

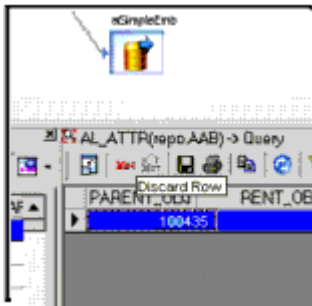
27.4.3.4 View Data pane

The View Data pane for lines uses the same tool bar and navigation options described for the View Data feature.

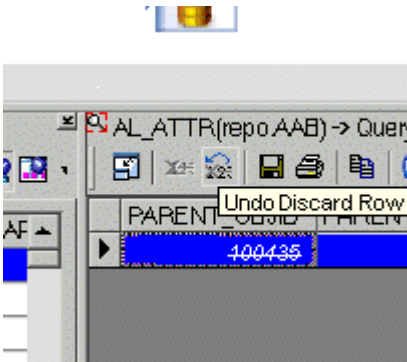
The following View Data pane options are unique to the interactive debugger:

- Allows you to view data that passes through lines.
- Displays (above the View Data tool bar) the names of objects to which a line connects using the format: `<TableName>(<DatastoreName>.<TableOwnerName>)-><QueryName>`.
- Displays data one row at a time by default.
- Provides the [All](#) check box which allows you to see more than one row of processed data.

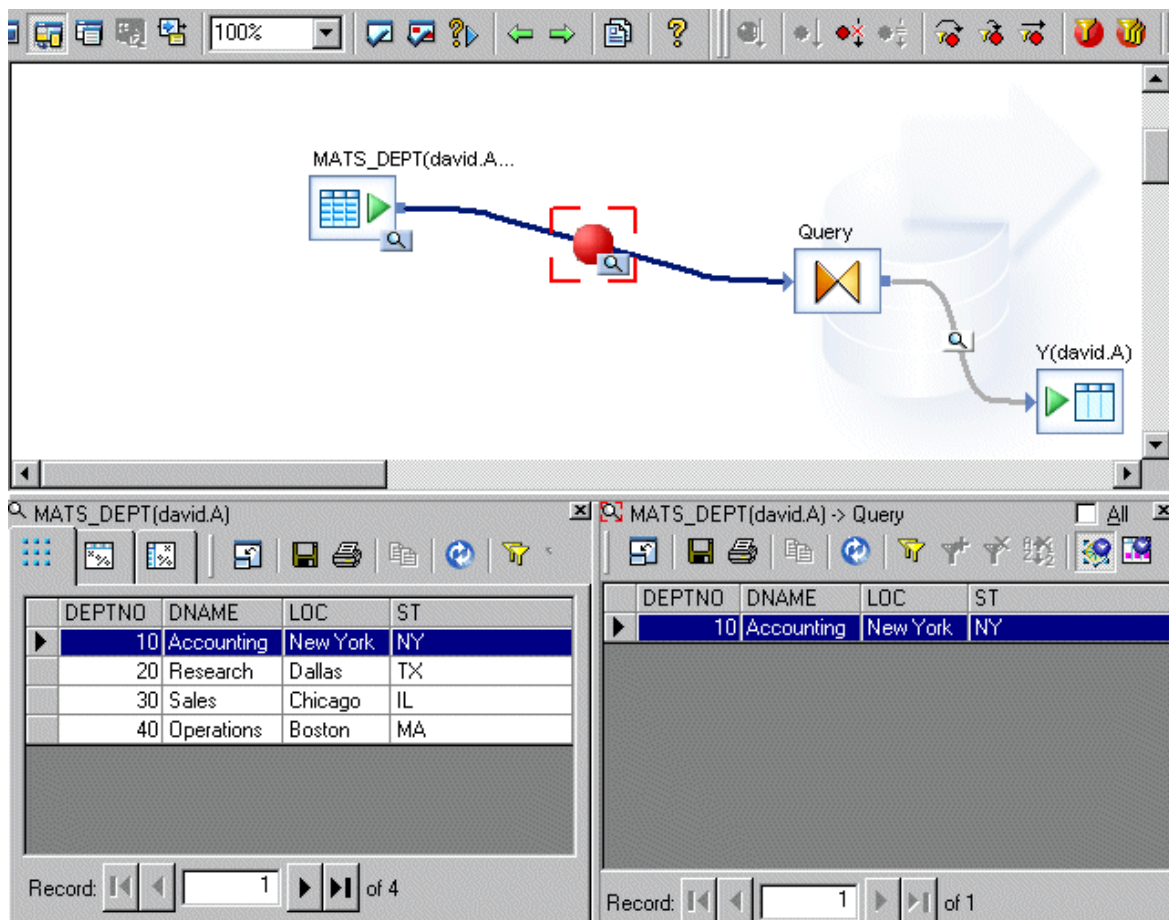
- Allows you to edit data in a cell.
For example, You might want to fix an error temporarily to continue with a debugger run. You can fix the job design later to eliminate the error permanently.
To edit cell data:
 - Deselect the [All](#) check box so that only has one row displayed.
 - Double-click a cell or right-click it and select [Edit cell](#).
- Uses a property called the [Data sample rate](#).
- Allows you to flag a row that you do not want the next transform to process.
To discard a row from the next step in a data flow process, select it and click [Discard Row](#).



Discarded row data appears in the strike-through style in the View Data pane (for example, 100345).
If you accidentally discard a row, you can undo the discard immediately afterwards. Select the discarded row and click [Undo Discard Row](#).



Alternatively, right-click a row and select either [Discard Row](#) or [Undo Discard Row](#) from the shortcut menu.
For example, if a source in a data flow has four rows and you set the Data sample rate to 2 when you start the debugger, it displays the first row processed at a pre-defined breakpoint.



If you use the [Get Next Row](#) option, then the next row at the same breakpoint is displayed.

If you want to see both rows, select the [All](#) check box on the upper-right corner of this pane. The row displayed at the bottom of the table is the last row processed.

At this point, you have viewed two rows that have passed through a line.

If you click [Get Next Row](#) again, only the last two rows processed are displayed because you set the sample size to 2.

Related Information

[Using View Data \[page 819\]](#)

27.4.3.5 Filters and Breakpoints window



You can manage interactive debugger filters and breakpoints using the Filters/Breakpoints window. You can open this window from the Debug menu or tool bar.

Lines that contain filters or breakpoints are listed in the far-left side of the Filters/Breakpoints window. To manage these, select the line(s) that you want to edit, select a command from the list and click Execute. You can also select a single line on the left and view/edit its filters and breakpoints on the right side of this window. When you are finished using the Filters/Breakpoints window, click OK.

27.4.4 Debug menu options and tool bar

Once you start the interactive debugger, you can access appropriate options from the Designer's [Debug](#) menu and tool bar.

Image	Option	Description	Key Commands
	Execute	Opens the Execution Properties window from which you can select job properties then execute a job outside the debug mode. Available when a job is active in the workspace.	F8
	Start debug	Opens the Debug Properties window from which you can select job properties then execute a job in debug mode (start the debugger). Other Designer operations are set to read-only until you stop the debugger. Available when a job is active in the workspace.	Ctrl+F8
	Stop debug	Stops a debug mode execution and exits the debugger. All Designer operations are reset to read/write.	Shift+F8
	Pause debug	Allows you to manually pause the debugger. You can use this option instead of a breakpoint.	None

Image	Option	Description	Key Commands
	Step over	Allows you to manually move to the next line in a data flow by stepping over a transform in the workspace. Use this option to see the first row in a data set after it is transformed. The workspace displays a red square on the line to indicate the path you are using. If the transform you step over has multiple outputs, the Designer provides a popup menu from which you can select the logic branch you want to take.	F10
	Get next row	Allows you to stay at the current breakpoint and view the next row of data in the data set.	F11
	Continue	Allows you to give control of the job back to the Designer - You use the Pause debug option. - Another breakpoint is encountered. - The job completes.	Ctrl+F10 . The debugger continues until:
	Show Filters/Breakpoints	Shows all filters and breakpoints that exist in a job. When not selected, all filters and breakpoints are hidden from view. This option is always available in the Designer.	None
	Set Filter/Breakpoints...	Opens a dialog from which you can set, remove, edit, enable or disable filters and breakpoints. You can also set conditions for breakpoints. From the workspace, you can right-click a line and select the same option from a short cut menu. Available when a data flow is active in the workspace.	F9

Image	Option	Description	Key Commands
	Filters/Breakpoints...	Opens a dialog with which you can manage multiple filters and breakpoints in a data flow. Also offers the same functionality as the Set Filters/Breakpoints window. This option is always available in the Designer.	Alt+F9
	Call Stack	Shows or hides the Call Stack window.	None
	Variables	Shows or hides the Debug Variables window.	
	Trace	Shows or hides the Trace window.	None None

27.4.5 Viewing data passed by transforms

To view the data passed by transforms, execute the job in debug mode.

27.4.5.1 Viewing data passed by transforms

1. In the project area, right-click a job and click [Start debug](#).
The Debug Properties window opens.
2. Clear the [Exit the debugger when the job is finished](#) check box.
3. You can enter a value in the [Data sample rate](#) text box or leave the default value, which is 500.
4. Click [OK](#).

27.4.5.2 Viewing sample data in debug mode

1. While still in debug mode after the job completes, in the project area, click the name of the data flow to view.
2. Click the View Data button displayed on a line in the data flow.
3. Navigate through the data to review it. When done, click the [Stop debug](#) button on the toolbar.

27.4.6 Push-down optimizer

When the software executes a job, it normally pushes down as many operations as possible to the source database to maximize performance. Because the interactive debugger requires a job execution, the following push-down rules apply:

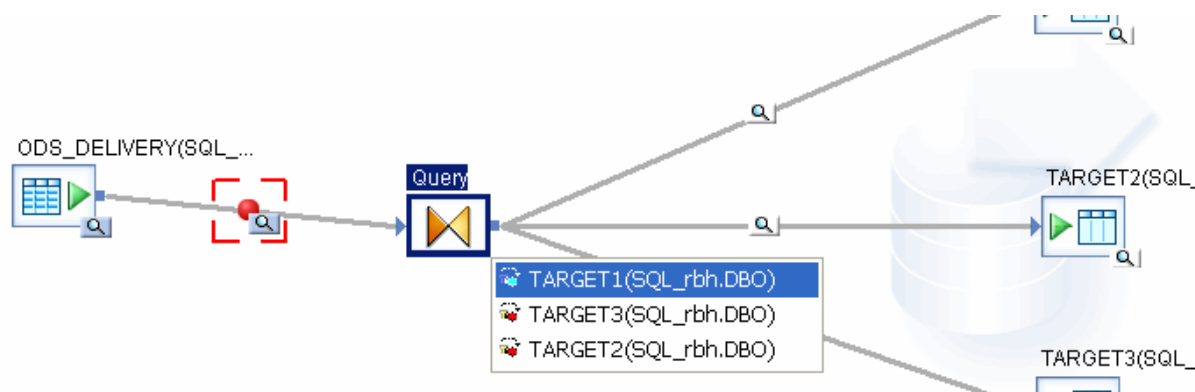
- **Query transforms**
The first transform after a source object in a data flow is optimized in the interactive debugger and pushed down to the source database if both objects meet the push-down criteria and if you have not placed a breakpoint on the line before the first transform.
- **Breakpoints**
The software does not push down any operations if you set a pre-defined breakpoint. Pre-defined breakpoints are breakpoints defined before you start the interactive debugger.
After the interactive debugger is started, if the first transform is pushed down, the line is disabled during the debugging session. You cannot place a breakpoint on this line and you cannot use the View Data pane.
- **Filters**
If the input of a pre-defined filter is a database source, it is pushed down. Pre-defined filters are interactive debugger filters defined before you start the interactive debugger.

Related Information

[Maximize push-down operations](#)

27.4.7 Limitations

- The interactive debugger can be used to examine data flows. Debug options are not available at the work flow level.
- The debugger cannot be used with ABAP data flows.
- All objects in a data flow must have a unique name. For example, if there are several outputs for a transform you can choose which path to use. If any of these objects have the same name, the result of your selection is unpredictable.



27.5 Comparing Objects

The software allows you to compare any two objects and their properties by using the Difference Viewer utility. You can compare:

- two different objects
- different versions of the same object
- an object in the local object library with its counterpart in the central object library

You can compare just the top-level objects, or you can include the object's dependents in the comparison.

Objects must be of the same type; for example, you can compare a job to another job or a custom function to another custom function, but you cannot compare a job to a data flow.

27.5.1 Comparing two different objects

1. In the local or central object library, right-click an object name.
2. From the shortcut menu, highlight *Compare*, and from the submenu, click one of the following options (availability depends on the object you selected):

Option	Description
<i>Object to central</i>	Compares the selected object to its counterpart in the central object library.
<i>Object with dependents to central</i>	Compares the selected object and its dependent objects to its counterpart in the central object library.
<i>Object to...</i>	Compares the selected object to another similar type of object.
<i>Object with dependents to...</i>	Compares the selected object and its dependents to another similar type of object.

The cursor changes to a target icon.

3. When you move the cursor over an object that is eligible for comparison, the target cursor changes color. Click on the desired object.

The *Difference Viewer* window opens in the workspace.

The window identifies changed items with a combination of icons, color, and background shading. Some of these properties are configurable.

Depending on the object type, the panes show items such as the object's properties and the properties of and connections (links) between its child objects.

27.5.2 Comparing two versions of the same object

If you are working in a multiuser environment and using a central object library, you can compare two objects that have different versions or labels.

1. In the central object library, right-click an object name, and from the shortcut menu click [Show History](#).
2. In the [History](#) window, Ctrl-click the two versions or labels you want to compare.
3. Click [Show Differences](#) or [Show Differences with Dependents](#).

The [Difference Viewer](#) window opens in the workspace.

4. Close the [History](#) window.

Related Information

[Viewing object history \[page 1024\]](#)

27.5.3 Overview of the Difference Viewer window

The first object you selected appears in the left pane of the window, and the second object appears on the right. Following each object name is its location.

The Difference Viewer window includes the following features:

- toolbar
- navigation bar
- status bar
- shortcut menu

Also, when a Difference Viewer window is active, the main designer window also contains a menu called Difference Viewer. The next section describes these features.

You can have multiple Difference Viewer windows open at a time in the workspace. To refresh a Difference Viewer window, press F5.

Expanding or collapsing any property set also expands or collapses the compared object's corresponding property set.

27.5.3.1 Toolbar

The toolbar includes the following buttons.

Navigation buttons

- First Difference (Alt+Home)
- Previous Difference (Alt+left arrow)
- Current Difference

- Next Difference (Alt+right arrow)
- Last Difference (Alt+End)

Filter buttons

- Enable filter(s)—Click to open the Filters dialog box.
 - *Hide non-executable elements*—Select this option to remove from view those elements that do not affect job execution.
 - *Hide identical elements*—Select this option to remove from view those elements that do not have differences.
- Disable filters—Removes all filters applied to the comparison.

Show levels

Show Level 1 shows only the objects you selected for comparison, Show Level 2 expands to the next level, etc. Show All Levels expands all levels of both trees.

Find (Ctrl+F)

Click *Find* to open a text search dialog box.

Open in new window

Click to open the currently active Difference Viewer in a separate window. You must close this window before continuing.

27.5.3.2 Navigation bar

The vertical navigation bar contains colored bars that represent each of the differences throughout the comparison. The colors correspond to those in the status bar for each difference. An arrow in the navigation bar indicates the difference that is currently highlighted in the panes. You can click on the navigation bar to select a difference (the cursor point will have a star on it). The purple brackets in the bar indicate the portion of the comparison that is currently in view in the panes.

Related Information

[Navigating through differences \[page 851\]](#)

27.5.3.3 Status bar

The status bar at the bottom of the window includes a key that illustrates the color scheme and icons that identify the differences between the two objects.

Icon	Difference	Description
	Deleted	The item does not appear in the object in the right pane.
	Changed	The differences between the items are highlighted in blue (the default) text.
	Inserted	The item has been added to the object in the right pane.
	Consolidated	This icon appears next to an item if items within it have differences. Expand the item by clicking its plus sign to view the differences

You can change the color of these icons by right-clicking in the Difference Viewer window and clicking Configuration.

The status bar also includes a reference for which difference is currently selected in the comparison (for example, the currently highlighted difference is 9 of 24 total differences in the comparison).

The status bar also indicates that there is at least one filter applied to this comparison.

Related Information

[Shortcut menu \[page 849\]](#)

27.5.3.4 Shortcut menu

Right-clicking in the conbody of the Difference Viewer window displays a shortcut menu that contains all the toolbar commands plus:

- View — Toggle to display or hide the status bar, navigation bar, or secondary toolbar (an additional toolbar that appears at the top of the window; you might find this useful if you have the Differences Viewer open in a separate window).
- Layout — Use to reposition the navigation bar.
- Configuration — Click to modify viewing options for elements with differences.

Related Information

[Changing the color scheme \[page 850\]](#)

27.5.3.4.1 Changing the color scheme

The status bar at the bottom of the Difference Viewer window shows the current color scheme being used to identify deleted, changed, inserted, or consolidated items in the comparison panes. You can customize this color scheme as follows.

1. Right-click in the conbody of the Difference Viewer window to display the shortcut toolbar.
2. Click Configuration to open the Configuration window.
3. Click a marker (Inserted, Deleted, Changed, or Consolidated) to change.
4. Click the Color sample to open the Color palette.
5. Click a Basic color or create a custom color.
6. Click [OK](#).
7. Click another marker to change it, or click [OK](#) to close the Configuration window.

27.5.3.4.2 Changing the background shading

Items with differences appear with a background default color of grey. You can customize this background.

1. Right-click in the conbody of the Difference Viewer window to display the shortcut toolbar.
2. Click Configuration to open the Configuration window.
3. Click a marker to change, or select the [Apply for all markers](#) check box.
4. Click the Background sample to open the Color palette.
5. Click a Basic color or create a custom color.
6. Click [OK](#).
7. To apply different background colors to different markers, click the marker to configure and repeat steps 4 through 6.
8. Click [OK](#) to close the Configuration window.

27.5.3.5 Difference Viewer menu

When a Difference Viewer window is active in the workspace, the main Designer window contains a menu called Difference Viewer. The menu contains the same commands as the toolbar.

27.5.4 Navigating through differences

The Difference Viewer window offers several options for navigating through differences.

You can navigate through the differences between the objects using the navigation buttons on the toolbar. For example, clicking the Next Difference button highlights the next item that differs in some way from the compared object. The item is marked with the appropriate icon and only the differing text appears highlighted in the color assigned to that type of difference.

You can also use the navigation bar. Select an item in either pane that has a difference. Note that an arrow appears next to the colored bar that corresponds to that item. You can click on these bars to jump to different places in the comparison, for example to view only inserted items (with a default color of green). The purple brackets in the bar indicate the portion of the comparison that is currently in view in the panes. Use the scroll bar in either pane to adjust the bracketed view.

For text-based items such as scripts, click the magnifying glass to view the text in a set of new panes that appear below the main object panes. Use the scroll bars for these panes to navigate within them. Click the magnifying glass (or any other item) to close the text panes.

27.6 Calculating column mappings

SAP Data Services can calculate information about target tables and columns and the sources used to populate them, for example for impact and lineage or auto documentation reports.

Calculating column mappings populates the internal ALVW_MAPPING view and the AL_COLMAP_NAMES table. The ALVW_MAPPING view provides current data to metadata reporting applications like Impact and Lineage Analysis. If you need to generate a report about a data flow that processes nested (NRDM) data, query the AL_COLMAP_NAMES table using a custom report.

Whenever a column mapping calculation is in progress, the Designer displays a status icon at the bottom right of the window. You can double-click this icon to cancel the process.

To calculate column mappings, you can:

- Enable the option in the Designer to automatically calculate column mappings.
- Execute the column mapping process manually from either the Designer or the Impact and Lineage Analysis application in the Management Console.

Related Information

[Reference Guide: Metadata in Repository Tables and Views, Storing nested column-mapping data](#)

27.6.1 Automatically calculating column mappings

To set the option to automatically calculate column mapping information, in the Designer select ► [Tools](#) ► [Options](#) ► [Designer](#) ► [General](#) ► [Automatically calculate column mappings](#) . This option is selected by default.

Note that if the Designer option [Automatically calculate column mappings](#) is cleared, any subsequent changes made to the data flow require that you manually recalculate the column mappings to ensure the ALVW_MAPPING view and the AL_COLMAP_NAMES table have the most current information.

27.6.2 Manually calculating column mappings

If the Designer option [Automatically calculate column mappings](#) is cleared and you want to generate reports, you can manually calculate the mappings. You can manually calculate column mappings at any time in either the Designer or the Management Console.

In the Designer, right-click in the object library and select ► [Repository](#) ► [Calculate column mappings](#) .

In the Management Console:

1. Select [Impact and Lineage Analysis](#).
2. Open the [Settings](#) control panel.
3. Click the [Refresh Usage Data](#) tab.
4. Select the Job Server that is associated with the repository you want to use.
5. Click [Calculate Column Mapping](#).

On the Impact and Lineage Analysis [Overview](#) tab, you can expand [Data Flow Column Mapping Calculation](#) to view a list of data flows and the calculation status of each. If the mapping calculation is complete, the [Status](#) indicator is checked.

27.7 Bypassing specific work flows and data flows

You can bypass the execution of a work flow or data flow during design time.

The Bypass attribute can help speed up the testing process when designing jobs by allowing you to run certain work flows or data flows in the job instead of having to run them all. Note that all objects within a bypassed data flow or work flow (for example, scripts) will also be bypassed.

i Note

You must create substitution parameters to use with the [Bypass](#) option. For example, in the [Substitution Parameter Editor](#) window you might create \$\$BYPASSEnable with a value of Yes and \$\$BYPASSDisable with a value of No (or any value other than Yes).

The following limitations apply:

- Bypassed data flows and work flows are ignored in Debug mode. Bypassing works only in design mode.

- ABAP data flows and embedded data flows are not supported.
- Bypass attributes are not supported during export and will be removed.

There are several ways to see if an object has been bypassed:

- After validation, a message displays in the [Warning](#) tab in the [Output](#) Window (for example, `Work Flow <work flow name> is bypassed.`). You can click on the message to go directly to the bypassed object.
- A message displays in the trace log.
- An icon (red circle with a line going through it) appears in the lower left corner of the data flow or work flow object in the Designer workspace.

i Note

This icon displays as long as the bypass attribute (enabled or disabled) is applied to an object.

Related Information

[Changing properties of a data flow \[page 275\]](#)

[Order of execution in work flows \[page 633\]](#)

[Creating a new substitution parameter \[page 724\]](#)

27.7.1 Bypassing a single data flow or work flow

You can enable the Bypass attribute for a single work flow or data flow.

1. Right-click on a work flow or data flow in the Designer workspace or Project Area and select [Properties](#).
2. In the [General](#) tab, select a substitution variable from the [Bypass](#) drop-down list.
3. Click [Apply](#).

27.7.2 Bypassing multiple data flows or work flows

You can enable the Bypass attribute for multiple work flows or data flows.

1. Select work flows and data flows in the Designer workspace.
2. Right-click and select [Bypass](#).
The [Set Bypass](#) window appears.
3. Select a substitution variable from the [Bypass](#) drop-down list.
4. Click [OK](#).

27.7.3 Disabling bypass

After you finish designing and testing your job, you should disable the Bypass attribute before moving the job to production mode.

To disable the Bypass attribute at the job level, right-click on a data flow or work flow in the Designer Project Area and select *Remove Bypass*.

To disable the Bypass attribute at the call level for multiple work flows or data flows, do the following:

1. Select work flows and data flows in the Designer workspace.
2. Right-click and select *Bypass*.

The *Set Bypass* window appears.

3. Select *{No Bypass}* from the *Bypass* drop-down list.
4. Click *OK*.

28 Recovery Mechanisms

Recovery mechanisms are available in SAP Data Services for batch jobs only.

Related Information

[Recovering from unsuccessful job execution \[page 855\]](#)

[Automatically recovering jobs \[page 856\]](#)

[Manually recovering jobs using status tables \[page 863\]](#)

[Processing data with problems \[page 864\]](#)

28.1 Recovering from unsuccessful job execution

If an SAP Data Services job does not complete properly, you must fix the problems that prevented the successful execution of the job and run the job again.

However, during the failed job execution, some data flows in the job may have completed and some tables may have been loaded, partially loaded, or altered. Therefore, you need to design your data movement jobs so that you can recover—that is, rerun the job and retrieve all the data without duplicate or missing data.

You can use various techniques to recover from unsuccessful job executions. This section discusses two techniques:

- Automatically recovering jobs — A software feature that allows you to run unsuccessful jobs in recovery mode.
- Manually recovering jobs using status tables — A design technique that allows you to rerun jobs without regard to partial results in a previous run.

You might need to use a combination of these techniques depending on the relationships between data flows in your application.

If you do not use these techniques, you might need to roll back changes manually from target tables if interruptions occur during job execution.

Related Information

[Automatically recovering jobs \[page 856\]](#)

[Manually recovering jobs using status tables \[page 863\]](#)

28.2 Automatically recovering jobs

With automatic recovery, the software records the result of each successfully completed step in a job. If a job fails, you can choose to run the job again in recovery mode. During recovery mode, the software retrieves the results for successfully completed steps and reruns uncompleted or failed steps under the same conditions as the original job. For recovery purposes, the software considers steps that raise exceptions as failed steps, even if the step is caught in a try/catch block.

28.2.1 Enabling automated recovery

To use the automatic recover feature, you must enable the feature during initial execution of a job. The software saves the results from successfully completed steps when the automatic recovery feature is enabled.

28.2.1.1 Running a job from Designer with recovery enabled

1. In the project area, select the job name.
2. Right-click and choose *Execute*.

The software prompts you to save any changes.

3. Make sure that the *Enable Recovery* option is selected in the *Execution Properties* window.

If this check box is not selected, the software does not record the results from the steps during the job and cannot recover the job if it fails. In that case, you must perform any recovery operations manually.

28.2.1.2 Running a job with recovery enabled from the Administrator

When you schedule or execute a job from the Administrator, select the *Enable Recovery* check box.

28.2.2 Marking recovery units

In some cases, steps in a work flow depend on each other and must be executed together. Because of the dependency, you should designate the work flow as a "recovery unit." When a work flow is a recovery unit, the entire work flow must complete successfully. If the work flow does not complete successfully, the software executes the entire work flow during recovery, even steps that executed successfully in prior work flow runs.

However, there are some exceptions to recovery unit processing. For example, when you specify that a work flow or a data flow should only execute once, a job will never re-execute that work flow or data flow after it completes successfully, except if that work flow or data flow is contained within a recovery unit work flow that re-executes and has not completed successfully elsewhere outside the recovery unit.

It is recommended that you not mark a work flow or data flow as *Execute only once* when the work flow or a parent work flow is a recovery unit.

Related Information

[Reference Guide: Data flow](#)

[Reference Guide: Work flow](#)

28.2.2.1 Specifying a work flow as a recovery unit

1. In the project area, select the work flow.
2. Right-click and choose *Properties*.
3. Select the *Recover as a unit* check box, then click *OK*.

During recovery, the software considers this work flow a unit. If the entire work flow completes successfully—that is, without an error—during a previous execution, then the software retrieves the results from the previous execution. If any step in the work flow did not complete successfully, then the entire work flow re-executes during recovery.

On the workspace diagram, the black "x" and green arrow symbol indicate that a work flow is a recovery unit.



28.2.3 Running in recovery mode

If a job with automated recovery enabled fails during execution, you can re-execute the job in recovery mode.

As with any job execution failure, you need to determine and remove the cause of the failure and rerun the job in recovery mode. If you need to make any changes to the job itself to correct the failure, you should not use automatic recovery. Instead, run the job as if it is a first run.

In recovery mode, the software executes the steps or recovery units that did not complete successfully in a previous execution—this includes steps that failed and steps that threw an exception but completed successfully such as those in a try/catch block. As in normal job execution, the software executes the steps in parallel if they are not connected in the work flow diagrams and in serial if they are connected.

Related Information

[Ensuring proper execution path \[page 858\]](#)

28.2.3.1 Running a job in recovery mode from Designer

1. In the project area, select the (failed) job name.
2. Right-click and choose *Execute*.

The software prompts you to save any objects that have unsaved changes.

3. Make sure that the *Recover from last failed execution* check box is selected.

This option is not available when a job has not yet been executed, when the previous run succeeded, or when recovery mode was disabled during previous run.

When you select Recover from last failed execution, the software retrieves the results from any steps that were previously executed successfully and executes or re-executes any other steps.

If you clear this option, the software runs this job anew, performing all steps.

When you schedule or execute a (failed) job from the Administrator, select the *Recover from last failed execution* check box.

28.2.4 Ensuring proper execution path

The automated recovery system requires that a job in recovery mode runs again exactly as it ran previously.

It is important that the recovery job run exactly as the previous run. If the job was allowed to run under changed conditions—suppose a `<sysdate>` function returns a new date to control what data is extracted—then the new data loaded into the targets will no longer match data successfully loaded into the target during the first execution of the job.

For example, suppose a daily update job running overnight successfully loads dimension tables in a warehouse. However, while the job is running, the database log overflows and stops the job from loading fact tables. The next day, the administrator truncates the log file and runs the job again in recovery mode. The recovery job does not reload the dimension tables because the original, failed run successfully loaded them.

To ensure that the fact tables are loaded with the data that corresponds properly to the data already loaded in the dimension tables, the recovery job must use the same extraction criteria that the original job used when loading the dimension tables. If the recovery job used new extraction criteria—such as basing data extraction on the current system date—the data in the fact tables would not correspond to the data previously extracted into the dimension tables.

In addition, if the recovery job uses new values, then the job execution may follow a completely different path through conditional steps or try/catch blocks.

If any global variable value is changed within a recovery as unit work flow, any downstream global variable reference is not guaranteed to have the updated value in the recovery mode execution. This will happen in the

recovery mode execution if the previous execution failure is not in the work flow that contains the variable change.

When recovery is enabled, the software stores results from the following types of steps:

- Work flows
- Batch data flows
- Script statements
- Custom functions (stateless type only)
- SQL function
- exec function
- get_env function
- rand function
- sysdate function
- systime function

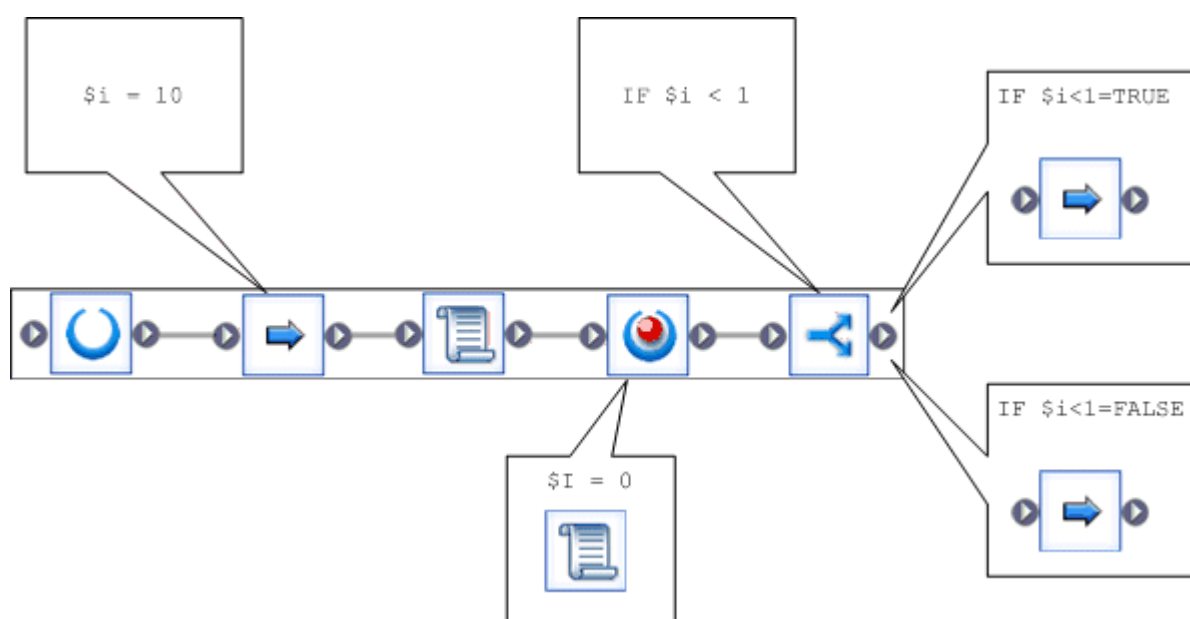
28.2.5 Using try/catch blocks with automatic recovery

SAP Data Services does not save the result of a try/catch block for reuse during recovery. If an exception is thrown inside a try/catch block, then during recovery the software executes the step that threw the exception and subsequent steps.

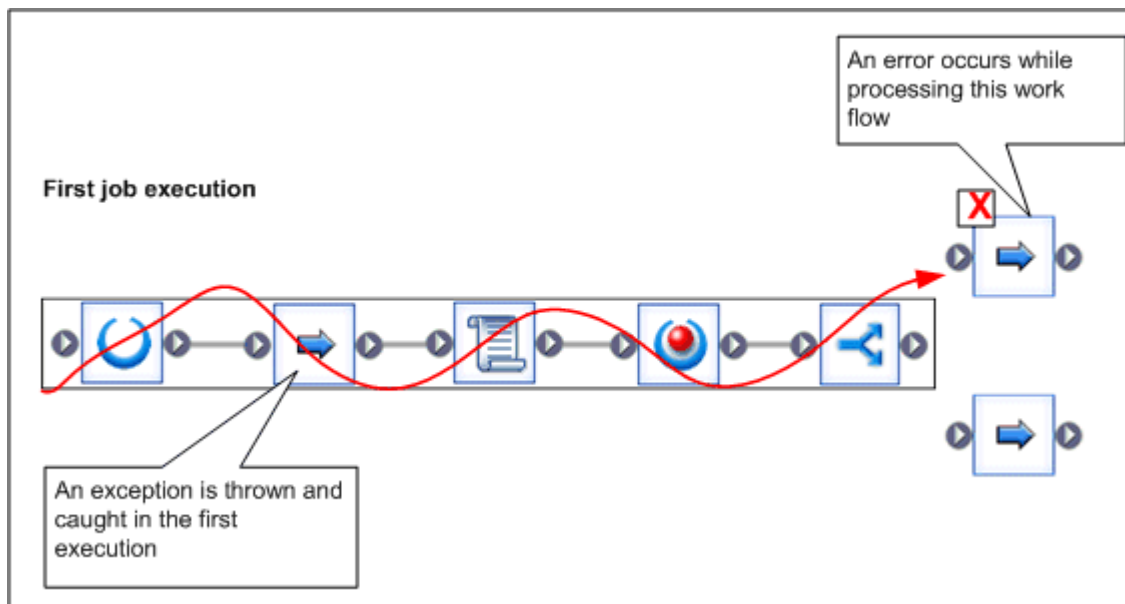
Because the execution path through the try/catch block might be different in the recovered job, using variables set in the try/catch block could alter the results during automatic recovery.

For example, suppose you create a job that defines a variable, `$i`, that you set within a try/catch block. If an exception occurs, you set an alternate value for `$i`. Subsequent steps are based on the value of `$i`.

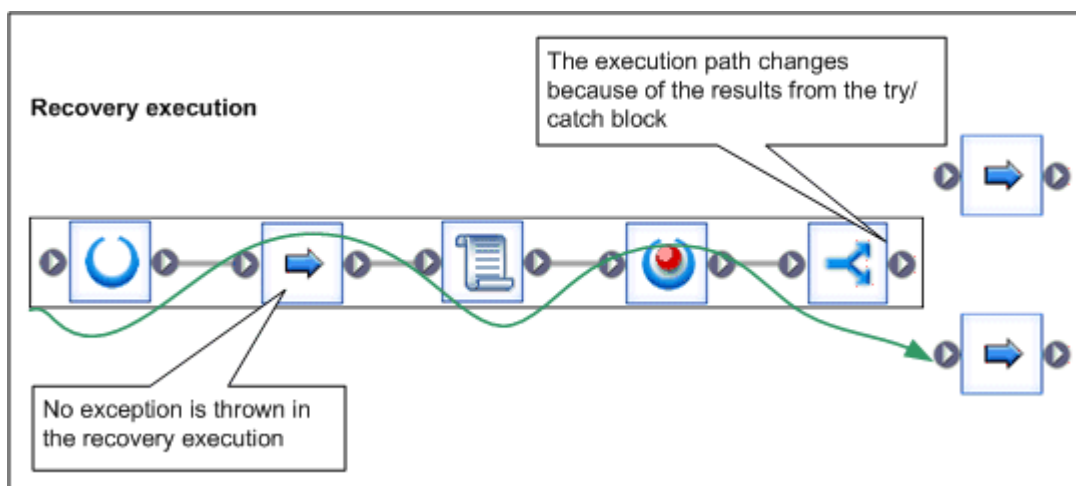
Job execution logic



During the first job execution, the first work flow contains an error that throws an exception, which is caught. However, the job fails in the subsequent work flow.



You fix the error and run the job in recovery mode. During the recovery execution, the first work flow no longer throws the exception. Thus the value of the variable, `<$i>`, is different, and the job selects a different subsequent work flow, producing different results.



To ensure proper results with automatic recovery when a job contains a try/catch block, do not use values set inside the try/catch block in any subsequent steps.

28.2.6 Ensuring that data is not duplicated in targets

Define work flows to allow jobs correct recovery. A data flow might be partially completed during an incomplete run. As a result, only some of the required rows could be inserted in a table. You do not want to insert duplicate rows during recovery when the data flow re-executes.

You can use several methods to ensure that you do not insert duplicate rows:

- Design the data flow to completely replace the target table during each execution
This technique can be optimal when the changes to the target table are numerous compared to the size of the table. You can use tuning techniques such as bulk loading options to improve overall performance.
- Set the auto correct load option for the target table
The auto correct load option checks the target table for existing rows before adding new rows to the table. Using the auto correct load option, however, can needlessly slow jobs executed in non-recovery mode. Consider this technique when the target table is large and the changes to the table are relatively few.
- Include a SQL command to execute before the table loads
Preload SQL commands can remove partial database updates that occur during incomplete execution of a step in a job. Typically, the preload SQL command deletes rows based on a variable that is set before the partial insertion step began.

28.2.7 Using preload SQL to allow re-executable data flows

To use preload SQL commands to remove partial database updates, tables must contain a field that allows you to tell when a row was inserted. Create a preload SQL command that deletes rows based on the value in that field.

For example, suppose a table contains a column that records the time stamp of any row insertion. You can create a script with a variable that records the current time stamp before any new rows are inserted. In the target table options, add a preload SQL command that deletes any rows with a time-date stamp greater than that recorded by the variable.



During initial execution, no rows match the deletion criteria. During recovery, the variable value is not reset. (The variable value is set in a script, which is executed successfully during the initial run.) The rows inserted during the previous, partial database load would match this criteria, and the preload SQL command would delete them.

To use preload SQL commands properly, you must define variables and pass them to data flows correctly.

28.2.7.1 Using preload SQL commands to ensure proper recovery

1. Determine appropriate values that you can use to track records inserted in your tables.

For example, if each row in a table is marked with the insertion time stamp, then you can use the value from the `sysdate()` function to determine when a row was added to that table.

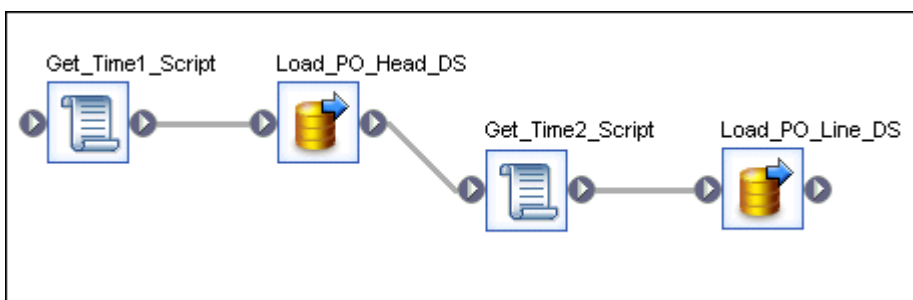
2. Create variables that can store the "tracking" values.

Variables are either job or work-flow specific. If a work flow is a recovery unit, create the "tracking" variables for that work flow at the job level; otherwise, create your tracking variables at the work flow level. Generally, you do not want tracking variables reset during recovery because when they reset, the preload SQL command will not work properly.

3. Create scripts that set the variables to the appropriate values.

Scripts are unique steps in jobs or work flows. You need to create a separate script that sets the required variables before each data flow or work flow that loads a table. If a work flow is a recovery unit, create the scripts for that work flow at the job level; otherwise, create the scripts at the work flow level.

4. Connect the scripts to the corresponding data flows or work flows.



5. Create parameters to pass the variable information from the job or work flow where you created the variable to the data flow that uses the tracking variable in the preload SQL command.
6. Insert appropriate preload SQL commands that remove any records inserted during earlier unsuccessful runs.

The preload SQL commands reference the parameter containing the tracking variable, deleting rows that were inserted after the variable was set.

For example, suppose the `<PO_ITEM>` table records the date-time stamp in the `<TIMESTAMP>` column. You created a variable `<$load_time>` that records the value from the `sysdate()` function before the load starts, and you passed that variable to the data flow that loads the `<PO_ITEM>` table in a parameter named

<\$load_time>. Then, your preload SQL command must delete any records in the table where the value in <TIMESTAMP> is larger than the value in <\$load_time>.

```
Delete from <PO_ITEM> where <TIMESTAMP> > [ <$load_time> ]
```

Related Information

[Defining a local variable \[page 710\]](#)

[Scripts \[page 645\]](#)

[Adding a parameter to a work flow or data flow \[page 707\]](#)

28.3 Manually recovering jobs using status tables

You can design your jobs and work flows so that you can manually recover from an unsuccessful run. A job designed for manual recovery must have certain characteristics:

- You can run the job repeatedly.
- The job implements special steps to recover data when a step did not complete successfully during a previous run.

You can use an execution status table to produce jobs that can be run multiple times without duplicating target rows. The table records a job's execution status. A "failure" value signals SAP Data Services to take a recovery execution path.

To implement a work flow with a recovery execution path:

- Define a flag that indicates when the work flow is running in recovery mode.
- Store the flag value in a status table.
- Check the flag value in the status table before executing a work flow to determine which path to execute in the work flow.
- Update the flag value when the work flow executes successfully.

For example, you could design a work flow that uses the auto correct load option when a previous run does not complete successfully. This work flow would have five steps, as illustrated:

1. Retrieve the flag value, which indicates the success or failure of the previous execution, from the status table. Store this value in a variable such as <\$recovery_needed> .
2. In a conditional, evaluate the <\$recovery_needed> variable.
3. If recovery is required, execute the recovery data flow recover_customer. This data flow loads the data using the auto correct load option.
4. If recovery is not required, execute the non-recovery data flow load_customer. This data flow loads the data without the auto correct load option.
5. Update the flag value in the status table to indicate successful execution.

Related Information

[Reference Guide: Target](#)

28.4 Processing data with problems

Jobs might not produce the results you expect because of problems with data. In some cases, the software is unable to insert a row. In other cases, the software might insert rows with missing information. You can design your data flows to anticipate and process these types of problems. For example, you might have a data flow write rows with missing information to a special file that you can inspect later.

This section describes mechanisms you can use to anticipate and process data problems. In particular, this section discusses three techniques:

- Using overflow files
- Filtering missing or bad values
- Handling facts with missing dimensions

28.4.1 Using overflow files

A row that cannot be inserted is a common data problem. Use the overflow file to process this type of data problem. When you specify an overflow file and the SAP Data Services cannot load a row into a table, the software writes the row to the overflow file instead. The trace log indicates the data flow in which the load failed and the location of the file.

For any table used as a target, you can set the option to use an overflow file in the [Options](#) tab. When you specify an overflow file, give a full path name to ensure that the software creates a unique file when more than one file is created in the same job. By default, the name of the overflow file is the target table name.

When you select the overflow file option, you choose what the software writes to the file about the rows that failed to load: either the data from the row or the SQL commands required to load the row. If you select data, you can use the software to read the data from the overflow file, cleanse it, and load it into the target table. If you select SQL commands, you can use the commands to load the target manually when the target is available.

There are many reasons for loading to fail, for example:

- Out of memory for the target
- Overflow column settings
- Duplicate key values

You can use the overflow information to identify invalid data in your source or problems introduced in the data movement. Every new run will overwrite the existing overflow file.

i Note

You cannot use overflow files when loading to a BW Transfer Structure.

28.4.2 Filtering missing or bad values

A missing or invalid value in the source data is another common data problem. Using queries in data flows, you can identify missing or invalid values in source data. You can also choose to include this data in the target or to disregard it.

For example, suppose you are extracting data from a source and you know that some phone numbers and customer names are missing. You can use a data flow to extract data from the source, load the data into a target, and filter the NULL values into a file for your inspection.

The data flow has five steps:

1. Extracts data from the source.
2. Selects the data set to load into the target and applies new keys. (It does this by using the Key_Generation function.)
3. Loads the data set into the target, using the bulk load option for best performance.
4. Uses the same data set for which new keys were generated in step 2, and select rows with missing customer names and phone numbers.
5. Writes the customer IDs for the rows with missing data to a file.

Now, suppose you do not want to load rows with missing customer names into your target. You can insert another query into the data flow to ensure that SAP Data Services does not insert incomplete rows into the target. The new query filters the rows with missing customer names before loading any rows into the target. The missing data query still collects those rows along with the rows containing missing phone numbers. In this version of the example, the Key_Generation transform adds keys for new rows before inserting the filtered data set into the target.

The data flow now has six steps.

1. Extracts data from the source.
2. Selects the data set to load into the target by filtering out rows with no customer name values.
3. Generates keys for rows with customer names.
4. Loads the valid data set (rows with customer names) into the target using the bulk load option for best performance.
5. Uses a separate query transform to select rows from the source that have no names or phones.
Note that the software does not load rows with missing customer names into the target; however, the software does load rows with missing phone numbers.
6. Writes the customer IDs for the rows with missing data to a file.

You could add more queries into the data flow to select additional missing or invalid values for later inspection.

28.4.3 Handling facts with missing dimensions

Another data problem occurs when SAP Data Services searches a dimension table and cannot find the values required to complete a fact table.

You can approach this problem in several ways:

- Leave the problem row out of the fact table.
Typically, this is not a good idea because analysis done on the facts will be missing the contribution from this row.

- Note the row that generated the error, but load the row into the target table anyway. You can mark the row as having an error, or pass the row information to an error file as in the examples from [Filtering missing or bad values \[page 865\]](#).
- Fix the problem programmatically. Depending on the data missing, you can insert a new row in the dimension table, add information from a secondary source, or use some other method of providing data outside of the normal, high-performance path.

28.5 Exchanging metadata

SAP Data Services enables you to export metadata into an XML file. After you create the file, you must manually import it into another tool.

Related Information

[Metadata exchange \[page 866\]](#)

28.5.1 Metadata exchange

You can exchange metadata between the software and third-party tools using XML files and the [Metadata Exchange](#) option.

The software supports two built-in metadata exchange formats:

- CWM 1.0 XML/XMI 1.1
CWM (the Common Warehouse Metamodel)— is a specification that enables easy interchange of data warehouse metadata between tools, platforms, and repositories in distributed heterogeneous environments.
- ERwin 4.x XML

The software can also use:

- MIMB (the *Meta Integration® Model Bridge*)
MIMB is a Windows stand-alone utility that converts metadata models among design tool formats. By using MIMB with the software, you can exchange metadata with all formats that MIMB supports. If MIMB is installed, the additional formats it supports are listed in the Metadata Exchange window.

28.5.1.1 Importing metadata files into the software

You can import metadata from ERwin Data Modeler 4.x XML into a datastore.

28.5.1.1.1 Importing metadata using Metadata Exchange

1. From the [Tools](#) menu, select [Metadata Exchange](#).
2. In the Metadata Exchange window, select [Import metadata from file](#).
3. In the [Metadata format](#) box, select `ERwin 4.x XML` from the list of available formats.
4. Specify the [Source file name](#) (enter directly or click [Browse](#) to search).
5. Select the [Target datastore name](#) from the list of datastores.
6. Click [OK](#) to complete the import.

28.5.1.2 Exporting metadata files from the software

You can export metadata into a file that other tools can read.

28.5.1.2.1 Exporting metadata using Metadata Exchange

1. From the [Tools](#) menu, select [Metadata Exchange](#).
2. In the Metadata Exchange window, select [Export Data Services metadata to file](#).
3. Select a [Metadata format](#) for the target from the list of available formats.

If you have MIMB installed and you select an MIMB-supported format, select the Visual check box to open the MIMB application when completing the export process.

If you do not select the Visual check box, the metadata is exported without opening the MIMB application. Using the MIMB application provides more configuration options for structuring the metadata in the exported file.

4. Specify the target file name (enter directly or click [Browse](#) to search).

When you search for the file, you open a typical browse window.

Find any of the following file formats/types:

Format	File type
DI CWM 1.0 XML/XMI 1.1	XML
DI ERwin 4.x XML	XML
MIMB format (only if installed)	All

After you select a file, click [Open](#).

5. Select the [Source datastore name](#) from the list of datastores.
6. Click [OK](#) to complete the export.

28.6 Loading Big Data file with recovery option

Loading big data file with recovery option is an extension of the existing recovery mechanism in Data Services. In earlier versions of Data Services, the recovery mechanism was only supported until the data flow layer. As a result, if the engine crashed at any data flow, the whole data flow had to be restarted from the beginning in recovery mode.

However, if there is big data inside of the failed dataflow, restarting the process will involve duplicating work of cleaning and restarting. This restarting situation becomes worse if there is a reason that the job needs to be stopped periodically (for example, due to a network disconnect, database timeout or machine reboot).

To avoid this situation, when reading a big source flat file into the database tables, you have the option to turn on the big data loading recovery feature. For example, when you load a large input file into a database, the job may take several days to finish. If the engine crashes or fails in the middle of a job, the job can be resumed from the last failed check point. The last failed check point includes the failed filename and the offset of the file that has been processed so far. The recovery feature avoids restarting of the data flow and ensures that loading progresses forward from where it failed.

28.6.1 Turning on the recovery option for Big Data loading

To use the automatic recovery feature for big data loading, you must enable the feature during the initial execution of a job. Consequently, when a job fails, it can resume from the last checked point and run forward instead of restarting.

To run a big data loading job from Designer with recovery enabled:

1. In the project area, select the job name.
2. Right-click and choose [Execute](#). The software prompts you to save any changes.
3. Make sure that the [Enable Recovery](#) option is selected in the [Execution Properties](#) window.

i Note

The [Enable Recovery](#) checkbox must be selected for the software to record the steps during the job. If the checkbox isn't selected and the job fails, you cannot recover the job. In this case, you have to perform any recovery operation manually and restart the job.

4. Check for the primary key in the target table of the data flow that contains the big data file source. If there is no primary key, then you have the option to either add the primary key column in the target table or generate the RowID column as the primary key column in the source file, and then pass it into the target tables.
 - a. To add the primary key in the target table: If the original schema of the target table does not have a primary key, turn on the [Use Input Keys](#) in the target table's option and make sure that the input to the target table has a primary key. For example, the row ID column that you enter in the source file can be passed to the target table as a primary key.
 - b. To generate Row ID column for the source file: Type a valid column name in the ROW ID input field located in the path [File Reader](#) > [General](#) .

i Note

A Row ID is automatically added into the schema as first column on the top. The column name is the name you typed, and the column Data Type is double.

A primary key column is necessary for fast recovery, as the engine uses auto correct loading for the last failed job. If the primary key column is not selected, then the auto correct loading will be very slow. To override this slow auto correct loading, you have the option to add the ROW ID column manually, if the target table does not have a primary key.

Data Services automatically turns on auto-correct loading for the recovery batch regardless of the target setting. Furthermore, for HANA, Netezza and Sybase IQ, the recovery batch auto-correct load uses bulk loading for better performance.

28.6.2 Limitations

There are some limitations that need to be met to allow a file to be loaded in recovery mode. The file along with the data flow and job contained in the file, needs to meet the following criteria:

- The file is a delimited file or fixed-width file
- The file has a row delimiter
- The job is running with [Enable recovery](#)
- The job is not a real-time job
- The job is not running in the Debugger
- The data flow is not within a continuous workflow
- The data flow is not audited
- The data flow is not an ABAP dataflow
- The data flow is not running in sub data flow mode
- The data flow does not contain any loader that does not connect to the file reader
- The data flow does not have the following transforms and functions:
 - Row_Generation
 - Date_Generation
 - gen_row_num
 - gen_row_num_by_group
 - CDC source
 - Target that is included in transaction
- The downstream of the file source does not have the following transforms:
 - Query transform that has an order by clause
 - Query transform that has a group by clause
 - Query transform that selects distinct rows
 - Query transform that has any aggregate function
 - Query transform that has a join in which file is an inner source

i Note

You can use the join rank to control the rank of the source relative to other tables and files, joined in a data flow. Sources with higher join ranks are joined before sources with lower join ranks.

- Hierarchy_Flattening transform
- Table_Comparison transform
- Reverse_Pivot transform

i Note

Reverse_Pivot transform on the downstream of the big file is allowed, only when "input data is grouped" option is checked.

- Validation transform
- Merge

i Note

Merge transform on the downstream of the big file is allowed, only when all inputs of the Merge transform are directly or indirectly from the file.

- Data Quality transform that uses a break key, such as Match transform
- Either the target table has a primary key or the input to the target table has a primary key and the target's 'Use input key' option is set to yes.

If any of the above conditions is not satisfied, recovery on the file is not enabled. However, there is no effect on the other source files or dataflows.

If the data flow contains multiple file sources, only one file source is chosen to run in recovery mode.

29 Changed Data capture

When you have a large amount of data to update regularly and a small amount of system down time for scheduled maintenance on a data warehouse, update data over time, or delta load. Two commonly used delta load methods are full refresh and changed-data capture (CDC).

[Full refresh \[page 871\]](#)

[Capture only changes \[page 872\]](#)

After an initial load is complete, you can choose to extract only new or modified data and update the target system.

[Source-based and target-based CDC \[page 872\]](#)

[Use CDC with Oracle sources \[page 874\]](#)

Use Oracle CDC to limit number of rows read regularly.

[Use CDC with Attunity mainframe sources \[page 890\]](#)

If your environment keeps large amounts of data current, the mainframe CDC feature is a simple solution to limiting the number of rows that must be read on a regular basis.

[Use CDC with SAP Replication Server \[page 898\]](#)

Data Services combined with SAP Replication Server's replication functionality provides two mechanisms to capture, transform, and propagate high volumes of data to a data warehouse in real time.

[Use CDC with Microsoft SQL Server databases \[page 914\]](#)

SAP Data Services utilizes three methods of change data capture in Microsoft SQL Server databases.

[Use CDC with timestamp-based sources \[page 935\]](#)

[Use CDC for targets \[page 950\]](#)

Source-based changed-data capture is almost always preferable to target-based capture for performance reasons.

29.1 Full refresh

Full refresh is easy to implement and easy to manage. This method ensures that no data will be overlooked or left out due to technical or programming errors. For an environment with a manageable amount of source data, full refresh is an easy method you can use to perform a delta load to a target system.

29.2 Capture only changes

After an initial load is complete, you can choose to extract only new or modified data and update the target system.

Identifying and loading only changed data is called changed-data capture (CDC). This includes only incremental data that has changed since the last refresh cycle. SAP Data Services acts as a mechanism to locate and extract only the incremental data that changed since the last refresh.

Improving performance and preserving history are the most important reasons for using changed-data capture.

- Performance improves because with less data to extract, transform, and load, the job typically takes less time.
- If the target system has to track the history of changes so that data can be correctly analyzed over time, the changed-data capture method can provide a record of these changes.

Example

If a customer moves from one sales region to another, simply updating the customer record to reflect the new region negatively affects any analysis by region over time because the purchases made by that customer before the move are attributed to the new region

29.3 Source-based and target-based CDC

Changed-data capture can be either source-based or target-based.

[Source-based CDC \[page 872\]](#)

[Target-based CDC \[page 873\]](#)

29.3.1 Source-based CDC

Source-based changed-data capture extracts only the changed rows from the source. It is sometimes called incremental extraction. This method is preferred because it improves performance by extracting the least number of rows.

SAP Data Services offers access to source-based changed data that various software vendors provide. The following table shows the supported data sources, changed-data capture products, and techniques.

Data Source	Products or techniques
Oracle 9i and later	Use Oracle's CDC packages to create and manage CDC tables. These packages make use of a publish and subscribe model. You can create a CDC datastore for Oracle sources using the Designer. You can also use the Designer to create CDC tables in Oracle, then import them for use in jobs.
Mainframe data sources (Adabas, DB2 UDB for z/OS, IMS, SQL/MP, VSAM, flat files) accessed with Attunity Connect	For mainframe data sources that use Attunity to connect to the software, you can use Attunity Streams 4.6.
Microsoft SQL Server databases	Use Microsoft SQL Replication Server to capture changed data from SQL Server databases.
Other sources	Use date and time fields to compare source-based changed-data capture job runs. This technique makes use of a creation and/or modification timestamp on every row. You can compare rows using the time of the last update as a reference. This method is called timestamp-based CDC.
SAP ODP sources	Define CDC filtering criteria when importing an ODP source.
i Note SAP extractors interface has been enhanced and replaced with OPD sources. Refer to the What's New Guide for Operational Data Provisioning [ODP] sources	
SAP Sybase Replication Server	Use SAP Sybase Replication Server to capture changed data.

Parent topic: [Source-based and target-based CDC \[page 872\]](#)

Related Information

[Target-based CDC \[page 873\]](#)

[Use CDC with Oracle sources \[page 874\]](#)

[Use CDC with Attunity mainframe sources \[page 890\]](#)

[Use CDC with Microsoft SQL Server databases \[page 914\]](#)

[Use CDC with timestamp-based sources \[page 935\]](#)

[Use CDC with SAP Replication Server \[page 898\]](#)

[Supplement for SAP: Connecting to SAP Applications, Capturing changed data, Importing for CDC](#)

29.3.2 Target-based CDC

Target-based changed-data capture extracts all the data from the source, but loads only the changed rows into the target.

Target-based changed-data capture is useful when you want to capture history but do not have the option to use source-based changed-data capture. The software offers table comparison to support this method.

Parent topic: [Source-based and target-based CDC \[page 872\]](#)

Related Information

[Source-based CDC \[page 872\]](#)

29.4 Use CDC with Oracle sources

Use Oracle CDC to limit number of rows read regularly.

If your environment must keep large amounts of data current, the Oracle Change Data Capture (CDC) feature is a simple solution to limiting the number or rows that the software reads on a regular basis. A source that reads only the most recent operations (INSERTS, UPDATES, DELETES), allows you to design smaller, faster delta loads.

[Overview of CDC for Oracle databases \[page 874\]](#)

[Set up Oracle CDC \[page 879\]](#)

System requirements so that your Oracle source database server tracks changes.

[Creating a CDC datastore for Oracle \[page 880\]](#)

[Import CDC data into tables \[page 881\]](#)

[Viewing an imported CDC table \[page 884\]](#)

[Configuring an Oracle CDC source table \[page 886\]](#)

[Creating a data flow with an Oracle CDC source \[page 888\]](#)

To use an Oracle CDC source, you use a Query transform to remove the Oracle control columns and the Map_CDC_Operation transform to interpret the control columns and take appropriate actions.

[Limitations \[page 890\]](#)

29.4.1 Overview of CDC for Oracle databases

With Oracle 9i or higher, SAP Data Services manages the CDC environment by accessing Oracle's Publish and Subscribe packages. Oracle publishes changed data from the original table to its CDC table.

The Designer allows you to create or import CDC tables and create subscriptions to access the data in the CDC table. Separate subscriptions allow each user to keep track of the last changed row that he or she accessed.

You can also enable check-points for subscriptions so that the software only reads the latest changes in the CDC table.

Oracle uses the following terms for Change Data Capture:

Term	Description
Change (CDC) table	A relational table that contains changed data that results from DML operations performed on a source table.
Change set	A group of CDC tables that are transactionally consistent. For example, SalesOrder and SalesItem tables should be in a change set to ensure that changes to an order and its line items are captured together.
Change source	The database that contains one or more change sets.
Publisher	The person who captures and publishes the changed data. The publisher is usually a database administrator (DBA) who creates and maintains the schema objects that make up the source database and staging database.
Publishing mode	Specifies when and how to capture the changed data. For details, see the following table of publishing modes.
Source database	The production database that contains the data that you extracted for your initial load. The source database contains the source tables.
Staging database	The database where the changed data is published. Depending on the publishing mode, the staging database can be the same as, or different from, the source database.
Subscriber	A user that can access the published data in the CDC tables.
Subscription	Controls access to the change data from one or more source tables within a single change set. A subscription contains one or more subscriber views.
Subscriber view	The changed data that the publisher has granted the subscriber access to use.

Oracle supports the following publishing modes:

Synchronous

- Data is captured using internal triggers on the source tables to store changes in CDC tables.
- Captured data is available in real-time.
- CDC tables must reside in the source database.

Considerations:

- Adds overhead to source database at capture time.
- Available in Oracle 9i and 10G.

Asynchronous HotLog

- Data is captured using redo or archive logs from the source database.
- Captured data is available in near real-time.
- A change set contains multiple CDC tables and must reside locally in the source database.

Considerations:

- Improves performance because data is captured offline.
- Available in Oracle 10G only.

Asynchronous AutoLog

- Data is captured using redo logs managed by log transport services that automate transfer from source database to staging database.

- Availability of captured data depends on the frequency of redo log switches on the source database.
- A change set contains multiple CDC tables and can be remote or local to the source database.

Considerations:

- Improves performance because data is captured offline.
- Available in Oracle 10G only.

Parent topic: [Use CDC with Oracle sources \[page 874\]](#)

Related Information

[Set up Oracle CDC \[page 879\]](#)

[Creating a CDC datastore for Oracle \[page 880\]](#)

[Import CDC data into tables \[page 881\]](#)

[Viewing an imported CDC table \[page 884\]](#)

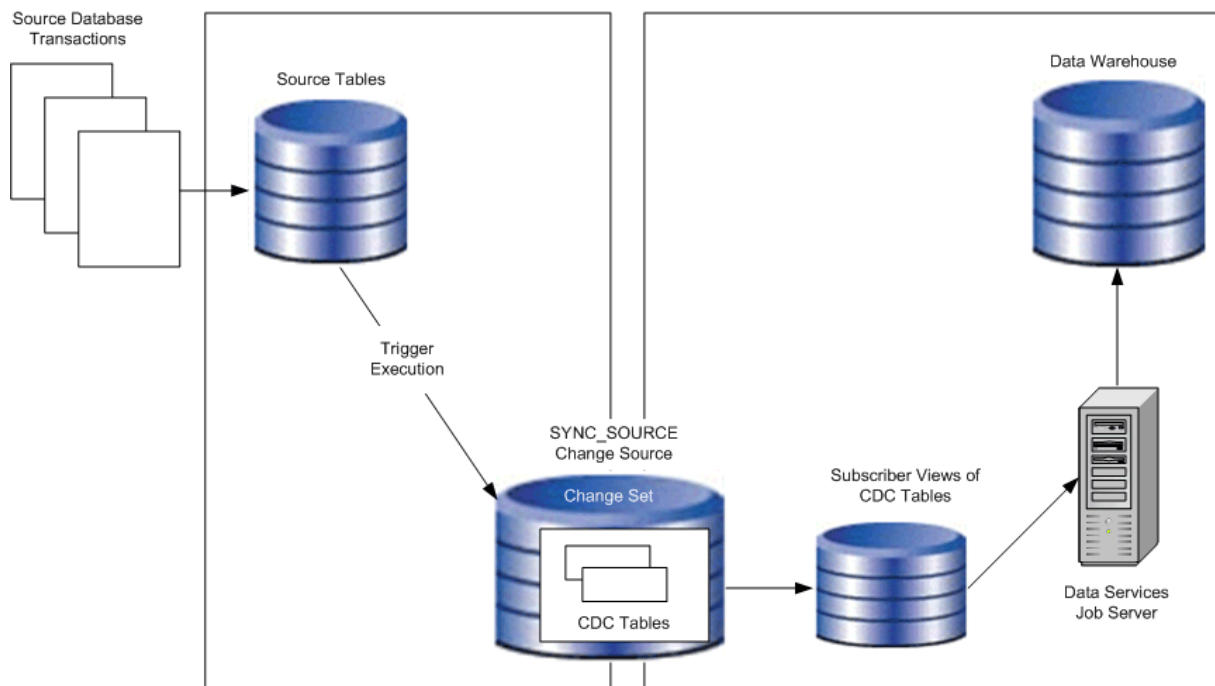
[Configuring an Oracle CDC source table \[page 886\]](#)

[Creating a data flow with an Oracle CDC source \[page 888\]](#)

[Limitations \[page 890\]](#)

29.4.1.1 Oracle CDC in synchronous mode

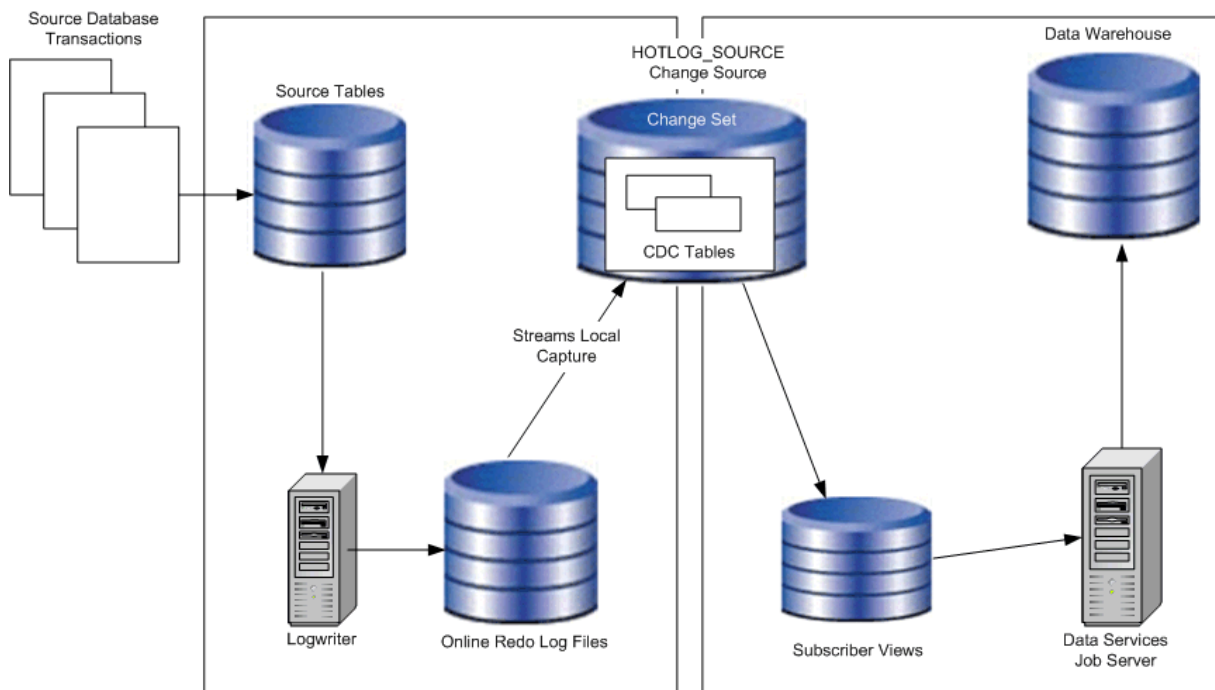
The following diagram shows how the changed data flows from Oracle CDC tables to SAP Data Services in synchronous mode.



When a transaction changes a source table, internal triggers capture the changed data and store it in the corresponding CDC table.

29.4.1.2 Oracle CDC in asynchronous HotLog mode

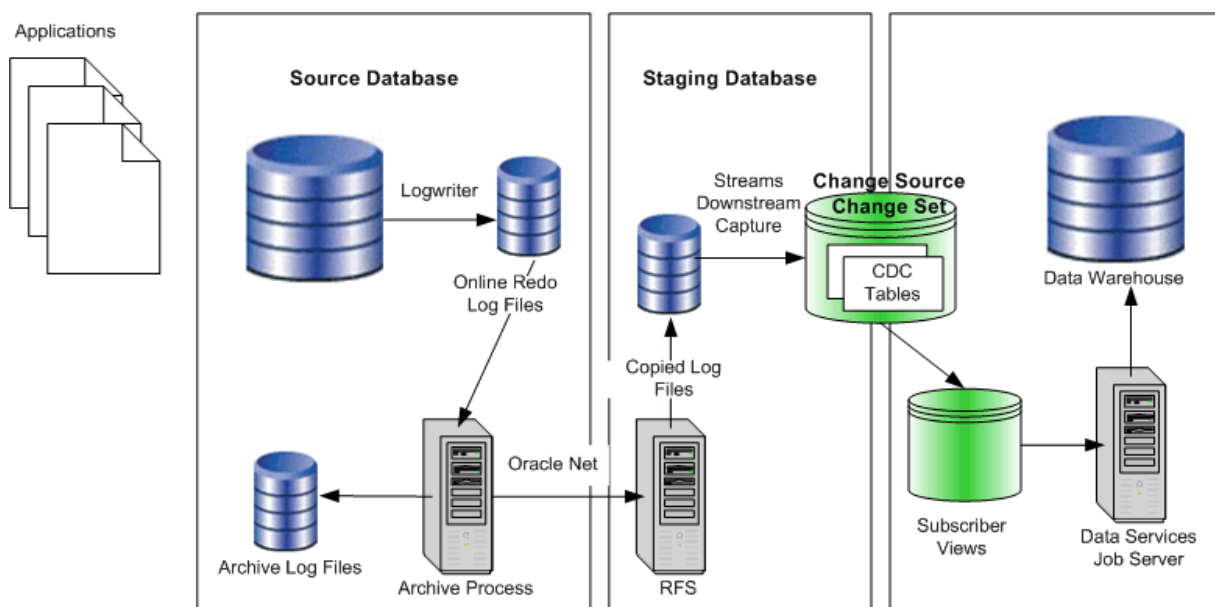
The following diagram shows how the changed data flows from Oracle CDC tables to SAP Data Services in asynchronous HotLog mode.



When a transaction changes a source table, the Logwriter records the changes in the Online Log Redo files. Oracle Streams processes automatically populate the CDC tables when transactions are committed.

29.4.1.3 Oracle CDC in asynchronous AutoLog mode

The following diagram shows how the changed data flows from Oracle CDC tables to SAP Data Services in asynchronous AutoLog mode.



When the log switches on the source database, Oracle archives the redo log file and copies the Online Log Redo files to the staging database. Oracle Streams processes populate the CDC tables from the copied log files.

i Note

The Oracle archive process requires uninterrupted connectivity through Oracle Net to send the redo log files to the remote file server (RFS).

29.4.2 Set up Oracle CDC

System requirements so that your Oracle source database server tracks changes.

The following table lists the system requirements on your Oracle source database server to track changes.

Requirement	Notes
Install Oracle's CDC packages	These packages are installed by default. However, if a CDC package needs to be re-installed, open Oracle's <code>Admin</code> directory, then find and run Oracle's SQL script <code>initcdc.sql</code>. • Synchronous CDC is available with Oracle Standard Edition and Enterprise Edition. • Asynchronous CDC is available with Oracle Enterprise Edition only.
Enable Java	
Set source table owner privileges	So that CDC tables can be created, purged, and dropped as needed.
Enable privileges for datastore owners	Privileges include: • <code>SELECT</code> privilege for CDC tables • <code>SELECT_CATALOG_ROLE</code> • <code>EXECUTE_CATALOG_ROLE</code> G
Enable Oracle's system triggers for synchronous CDC	

Requirement	Notes
For asynchronous AutoLog CDC	• The source database DBA must build a LogMiner data dictionary to enable the log transport services to send this data dictionary to the staging database. Oracle automatically updates the data dictionary with any source table DDL operations that occur during CDC to keep the staging tables consistent with the source tables. • The source database DBA must also obtain the SCN value of the data dictionary build. If you will use the Designer to create CDC tables, you need to specify the SCN in the wizard. • The publisher (usually the source database DBA) must configure log transport services to copy the redo log files from the source database system to the staging database system and to automatically register the redo log files.

Parent topic: [Use CDC with Oracle sources \[page 874\]](#)

Related Information

[Overview of CDC for Oracle databases \[page 874\]](#)

[Creating a CDC datastore for Oracle \[page 880\]](#)

[Import CDC data into tables \[page 881\]](#)

[Viewing an imported CDC table \[page 884\]](#)

[Configuring an Oracle CDC source table \[page 886\]](#)

[Creating a data flow with an Oracle CDC source \[page 888\]](#)

[Limitations \[page 890\]](#)

[Creating Oracle CDC tables in the software \[page 882\]](#)

29.4.3 Creating a CDC datastore for Oracle

To access CDC tables, create a CDC datastore using the Designer. A CDC datastore is a read-only datastore that can only access tables. Like other datastores, you can create, edit, and access a CDC datastore from the [Datastores](#) tab of the object library.

1. Create a database datastore with the [Database Type](#) option set to `Oracle`.
2. Select the [CDC](#) check box.
3. Select an [Oracle version](#).

The Designer only allows you to select the Oracle versions that support CDC packages.

4. Specify the name of your staging database (the change source database where the changed data is published) in *Connection name*.
5. Enter the *User* and *Password* for your staging database and click *OK*.

You can use this datastore to browse and import CDC tables.

Task overview: [Use CDC with Oracle sources \[page 874\]](#)

Related Information

[Overview of CDC for Oracle databases \[page 874\]](#)

[Set up Oracle CDC \[page 879\]](#)

[Import CDC data into tables \[page 881\]](#)

[Viewing an imported CDC table \[page 884\]](#)

[Configuring an Oracle CDC source table \[page 886\]](#)

[Creating a data flow with an Oracle CDC source \[page 888\]](#)

[Limitations \[page 890\]](#)

29.4.4 Import CDC data into tables

First, create one CDC table in Oracle for every source table that you want to read. Then you can import that CDC table using SAP Data Services. Create CDC tables in one of the following ways:

- Use an Oracle utility
- Use Data Services Designer

Parent topic: [Use CDC with Oracle sources \[page 874\]](#)

Related Information

[Overview of CDC for Oracle databases \[page 874\]](#)

[Set up Oracle CDC \[page 879\]](#)

[Creating a CDC datastore for Oracle \[page 880\]](#)

[Viewing an imported CDC table \[page 884\]](#)

[Configuring an Oracle CDC source table \[page 886\]](#)

[Creating a data flow with an Oracle CDC source \[page 888\]](#)

[Limitations \[page 890\]](#)

29.4.4.1 Using existing Oracle CDC tables

1. Import an Oracle CDC table by right-clicking the CDC datastore name in the object library and selecting [Open](#), [Import by Name](#), or [Search](#).

If you select Open, you can browse the datastore for existing CDC tables using the Datastore Explorer.

2. When you find the table that you want to import, right-click it and select [Import](#).

29.4.4.2 Creating Oracle CDC tables in the software

The software provides the ability to create Oracle CDC tables for all publishing modes:

- Synchronous CDC
 - Asynchronous HotLog CDC
 - Asynchronous AutoLog CDC
1. In the object library, right-click a CDC datastore and select [Open](#).
 2. In the Datastore Explorer, right-click the white space in the [External Metadata](#) section, and select [New](#).

The New CDC table wizard opens. This wizard allows you to add a CDC table.

i Note

If the Datastore Explorer opens and no CDC tables exist in your datastore, this wizard opens automatically.

3. Select the publishing mode on the first page of the wizard.

If your source database is Oracle 9i, you can only select the Synchronous mode. The Asynchronous modes are disabled.

If your source database is Oracle 10G, the wizard selects the Asynchronous HotLog mode by default.

If your source database uses Asynchronous AutoLog publishing mode, select Asynchronous AutoLog and provide the following source database connection information:

Field	Description
Connection name	The name of the database where the Change Source resides. Use the service name of the Oracle Net service configuration.
User Name	The user name for the source database DBA.
Password	The password for the Change Source user.

4. Click [Next](#). The second page of the wizard appears.
5. Specify the source table information in the second page of the wizard.
 - a. Click the [Search](#) button to see a list of non-CDC external tables available in this datastore. To filter a search, enter values for a table Owner and/or Name. You can use a wild-card character (%) to perform pattern matching for Name or Owner values.
 - b. (Optional) Select [Generate before-images](#) if you want to track before- and after-images in the new CDC table.

- c. Click a name in the list of returned tables and click [Next](#) to create a CDC table using the selected table as a source table.
6. Specify the [CDC table owner](#) for the new CDC table.

By default, the owner name of the new CDC table is the owner name of the datastore. The source table owner name is also displayed in the CDC table owner list box. If the owner name you want to use is not in the list, enter a different owner name.
7. Specify the [CDC table name](#) for the new CDC table.

By default, the software generates a table name using the following convention: CDC__SourceTableName.
8. By default, all columns are selected. Specify which columns to include or exclude from the CDC table in one of the following ways: Either remove the check mark from the box next to the name of each column that you want to exclude, or click [Unselect All](#) and place a check mark next to the name of each column that you want to include.
9. For synchronous publishing mode:
 - a. Click [Finish](#). The Designer connects to the Oracle instance, creates the CDC table on the Oracle server, and imports the table's metadata into the repository. All tables that the software imports through a CDC datastore contain a column that indicates which operation to perform for each row. For an Oracle CDC table, this column is called Operation\$. In addition to this column, Oracle adds other columns when it creates a CDC table. These columns all use a dollar sign as a suffix.
 - b. Click [OK](#) on the information dialog. This dialog confirms that Oracle created a new CDC table, then imported it successfully into the software.
10. For asynchronous (HotLog or AutoLog) publishing mode, click [Next](#).
11. For asynchronous HotLog publishing mode, specify the change set information in the fourth page of the wizard.
 - a. If you would like to add this change table to an existing change set to keep the changes transactionally consistent with the tables in the change set, select a name from the drop-down list for [Change set name](#). Alternatively, you can create a new change set by typing in the name.
 - b. Select [Stop capture on DDL](#) if a DDL error occurs and you do not want to capture data.
 - c. Select [Define retention period](#) to enable the [Begin Date](#) and [End Date](#) text boxes.
 - d. Click [Finish](#).

The Designer connects to the Oracle instance, creates the CDC table on the Oracle server, and imports the table's metadata into the software's repository. All tables that the software imports through a CDC datastore contain a column that indicates which operation to perform for each row. For an Oracle CDC table, this column is called Operation\$. In addition to this column, Oracle adds other columns when it creates a CDC table. These columns all use a dollar sign as a suffix.
12. For asynchronous AutoLog publishing mode, specify the change set and change source information in the fourth page of the wizard.
 - a. If you would like to add this change table to an existing change set to keep the changes transactionally consistent with the tables in the change set, select a name from the drop-down list for [Change set name](#). Alternatively, you can create a new change set by typing in the name.
 - b. If you would like to add this change table to an existing change source, select a name from the drop-down list for [Change source name](#).
 - c. If you want to create a new change source, type the name of the CDC change source and the name of the source database. You can obtain this name from the source database Global_Name table SCN value of the data dictionary build.
 - d. Select [Stop capture on DDL](#) if a DDL error occurs during data capture and you do not want to capture data.

- e. Select *Define retention period* to enable the *Begin Date* and *End Date* text boxes.
- f. Click *Finish*.

The Designer connects to the Oracle staging database, creates the CDC table on the change source, and imports the table's metadata into the software's repository. All tables that the software imports through a CDC datastore contain a column that indicates which operation to perform for each row. For an Oracle CDC table, this column is called *Operation\$*. In addition to this column, Oracle adds other columns when it creates a CDC table. These columns all use a dollar sign as a suffix.

Related Information

[Using before-images \[page 887\]](#)

29.4.5 Viewing an imported CDC table

To view an imported CDC table:

1. Find your CDC datastore in the object library.
2. Expand the *Tables* folder.
3. Double-click a table name or right-click and select *Open*.

When the software imports a CDC table, it also adds two columns to the table's schema:

DI_SEQUENCE_NUMBER with the data type integer and *DI_OPERATION_TYPE* with the data type *varchar(1)*.

An imported Oracle CDC table schema looks like the following:

	Description
CDC_ORDERS	
DI_SEQUENCE_NUMBER	DI-generated integer for sequencing
DI_OPERATION_TYPE	DI-generated row operation type
OPERATION\$	Oracle CDC table columns
CSCN\$	
COMMIT_TIMESTAMP\$	
RSID\$	
USERNAME\$	
TIMESTAMP\$	
ORDER_ID	Oracle source columns
ORDER_DATE	
CUSTOMER_ID	
ORDER_TOTAL	
SALES_REP_ID	
PROMOTION_ID	

This example has eight control columns added to the original table:

- Two generated by the software
 - `DI_SEQUENCE_NUMBER` - Contains an integer for sequencing.
 - `DI_OPERATION_TYPE` - Contains the row operation type.
- Six Oracle control columns: `OPERATION$`, `CSCN$`, `COMMIT_TIMESTAMP$`, `RSID$`, `USERNAME$`, `TIMESTAMP$`

i Note

The Oracle control columns vary, depending on the options that were selected when the CDC table is created. All Oracle control columns end with a dollar sign (\$).

Task overview: [Use CDC with Oracle sources \[page 874\]](#)

Related Information

[Overview of CDC for Oracle databases \[page 874\]](#)

[Set up Oracle CDC \[page 879\]](#)

[Creating a CDC datastore for Oracle \[page 880\]](#)

[Import CDC data into tables \[page 881\]](#)

[Configuring an Oracle CDC source table \[page 886\]](#)

[Creating a data flow with an Oracle CDC source \[page 888\]](#)

[Limitations \[page 890\]](#)

[The `DI\_SEQUENCE\_NUMBER` column \[page 885\]](#)

[The `DI\_OPERATION\_TYPE` column \[page 886\]](#)

29.4.5.1 The `DI_SEQUENCE_NUMBER` column

The `DI_SEQUENCE_NUMBER` column starts with zero at the beginning of each extraction. This field increments by one each time the software reads a row except when it encounters a pair of before- and after-images for an UPDATE operation. Both the before- and after-images receive the same sequence number. This sequencing column provides a way to collate image pairs if they are separated as a result of the data flow design.

Related Information

[Using before-images \[page 887\]](#)

29.4.5.2 The DI_OPERATION_TYPE column

The possible values for the `DI_OPERATION_TYPE` column are:

- I for INSERT
- D for DELETE
- B for before-image of an UPDATE
- U for after-image of an UPDATE

When the software reads rows from Oracle, it checks the values in column `Operation$` and translates them to the software values in the `DI_OPERATION_TYPE` column.

Operation\$	DI_OPERATION_TYPE
I	I
D	D
UO, UU	B
UN	U

29.4.6 Configuring an Oracle CDC source table

When you drag a CDC datastore table into a data flow, it automatically becomes a source object.

1. Drag a CDC datastore table into a data flow.
2. Click the name of this source object to open its Source Table Editor.
3. Click the CDC Options tab.
4. Specify a value for the [CDC subscription name](#).

For more information, see the *Reference Guide*.

Task overview: [Use CDC with Oracle sources \[page 874\]](#)

Related Information

[Overview of CDC for Oracle databases \[page 874\]](#)

[Set up Oracle CDC \[page 879\]](#)

[Creating a CDC datastore for Oracle \[page 880\]](#)

[Import CDC data into tables \[page 881\]](#)

[Viewing an imported CDC table \[page 884\]](#)

[Creating a data flow with an Oracle CDC source \[page 888\]](#)

[Limitations \[page 890\]](#)

29.4.6.1 Using check-points

When a job in SAP Data Services runs with check-pointing enabled, software uses the source table's subscription name to read the most recent set of appended rows. If you do not enable check-pointing, then the job reads all the rows in the table and increases processing time.

To use check-points, enter a name in the *CDC Subscription* name box on the Source Table Editor and select the *Enable check-point* option.

i Note

To avoid data corruption problems, do not reuse data flows that use CDC datastores because each time a source table extracts data it uses the same subscription name. This means that identical jobs, depending upon when they run, can get different results and leave check-points in different locations in the table. When you migrate CDC jobs from test to production, for example, a best practice scenario would be to change the subscription name for the production job so that the test job, if ever runs again, will not affect the production job's results.

29.4.6.2 Using before-images

If you want to retrieve the before-images of UPDATE rows, prior to when the update operation is applied to the target, the software can expand the UPDATE row into two rows: one row for the before-image of the update, and one row for the after-image of the update. The before image of an update row is the image of the row before the row is changed, and the after image of an update row refers to the image of the row after the change is applied.

The default behavior is that a CDC reader retrieves after-images only. By not retrieving before-images, fewer rows pass through the engine which allows the job to execute in less time.

You can use before-images to:

- Update primary keys

i Note

Under most circumstances, when source tables are updated, their primary keys do not need to be updated.

- Calculate change logic between data in columns
For example, you can calculate the difference between an employee's new and old salary by looking at the difference between the values in salary fields.

29.4.6.2.1 Capturing before-images for update rows

1. At CDC table creation time, make sure the Oracle CDC table is also setup to retrieve full before-images.
If you create an Oracle CDC table using the Designer, you can select the *Generate before-images* to do this.
2. Select *Get before-images for each update row* in the CDC table's source editor.

If the underlying, CDC table is not set-up properly, enabling the Get before-images for each update row option has no effect.

Once you select the *Get before-images for each update row* option, for every update, the software processes two rows. In addition to the performance impact of this data volume increase, the before- and after-image pairs may be separated or lost depending on the design of your data flow. This would cause data integrity issues.

The Map_CDC_Operation transform can resolve problems, but undesirable results may still occur due to programming errors. When using functions and transforms that re-order, re-direct, eliminate, and multiply the number of rows in a data flow (for example, due to the use of the group by or order by clauses in a query) be aware of the possible impact to targets.

29.4.7 Creating a data flow with an Oracle CDC source

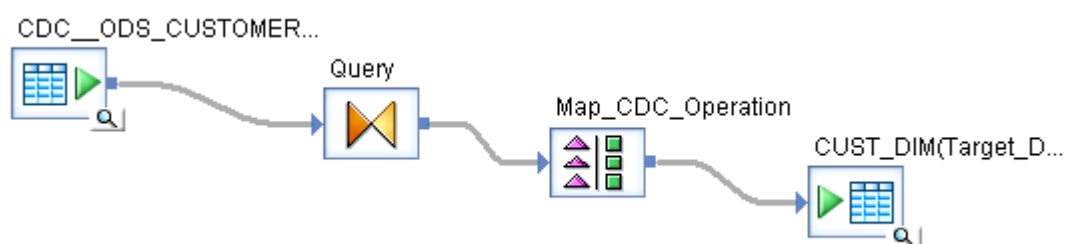
To use an Oracle CDC source, you use a Query transform to remove the Oracle control columns and the Map_CDC_Operation transform to interpret the control columns and take appropriate actions.

1. In the *Designer Object Library* pane, drag the Oracle CDC table, Query, and Map_CDC_Operation transforms to the data flow workspace area.

i Note

A data flow can contain only one CDC source.

2. Configure the CDC table.
3. Add the appropriate target table and connect the objects.



4. In the *Project Area* pane, double-click a transform. The *Query Editor* opens, displaying the transform.
5. Map the *Data Services* control columns and the source table columns that you want in your target table. The Map_CDC_Operation transform uses the values in the column in the *Row Operation Column* box to perform the appropriate operation on the source row for the target table. For an Oracle CDC source table, the `DI_OPERATION_TYPE` column is automatically selected as the Row operation column.

The operations can be `INSERT`, `DELETE`, or `UPDATE`. For example, if the operation is `DELETE`, the corresponding row is deleted from the target table.

Task overview: [Use CDC with Oracle sources \[page 874\]](#)

Related Information

[Overview of CDC for Oracle databases \[page 874\]](#)
[Set up Oracle CDC \[page 879\]](#)
[Creating a CDC datastore for Oracle \[page 880\]](#)
[Import CDC data into tables \[page 881\]](#)
[Viewing an imported CDC table \[page 884\]](#)
[Configuring an Oracle CDC source table \[page 886\]](#)
[Limitations \[page 890\]](#)
[Configuring an Oracle CDC source table \[page 886\]](#)

29.4.8 Maintaining CDC tables and subscriptions

29.4.8.1 Purging CDC tables

Periodically purge CDC tables so they do not grow indefinitely.

i Note

The software does not provide this functionality. Refer to your Oracle documentation for how to purge data that is no longer being used by any subscribers.

29.4.8.2 Dropping Oracle CDC subscriptions or tables

Oracle's purge facility does not purge any data that has not been read by all subscriptions. As a result, it is a good practice to drop any subscriptions that are no longer needed. You can drop Oracle CDC tables and their subscriptions from the *Datastore Explorer* window in the Designer.

1. In the *Local Object Library* pane, right-click a CDC datastore, and click *Open*.
2. In the *Datastore Explorer* window, click *Repository Metadata*.
3. Right-click a table, point to *CDC maintenance*, and do one of the following:
 - Point to *Drop subscription*.
 - *Drop Subscription* opens the list of subscriptions you created in the software for the selected table. Oracle subscriptions are associated with these subscription names. Select each subscription name to drop it from Oracle and delete it from the repository.
 - Click *Drop table*.
 - This option drops the Oracle CDC table and also deletes it from the repository.

29.4.9 Limitations

The following limitations exist when using CDC with Oracle sources:

- You cannot use the following transforms and functions with a source table imported with a CDC datastore because of the existence of the SAP Data Services generated columns for CDC tables. The software cannot compare or search these columns.
 - Table_Comparison, Key_Generation, and SQL transforms
 - All database functions, such as `lookup`, `lookup_ext`, `key_generation`, `sql`, and `total_rows`
- You can only create one CDC source in a data flow.
- Oracle CDC captures DML statements, including `INSERT`, `DELETE`, and `UPDATE`. However, Oracle CDC does not support the following operations because they disable all database triggers:
 - Direct-path `INSERT` statements
 - The `multi_table_insert` statement in parallel DML mode
- If you are using check-pointing and running your job in recovery mode, the recovered job will begin to review the job at the start of the CDC table. Check-points are ignored.

Parent topic: [Use CDC with Oracle sources \[page 874\]](#)

Related Information

[Overview of CDC for Oracle databases \[page 874\]](#)

[Set up Oracle CDC \[page 879\]](#)

[Creating a CDC datastore for Oracle \[page 880\]](#)

[Import CDC data into tables \[page 881\]](#)

[Viewing an imported CDC table \[page 884\]](#)

[Configuring an Oracle CDC source table \[page 886\]](#)

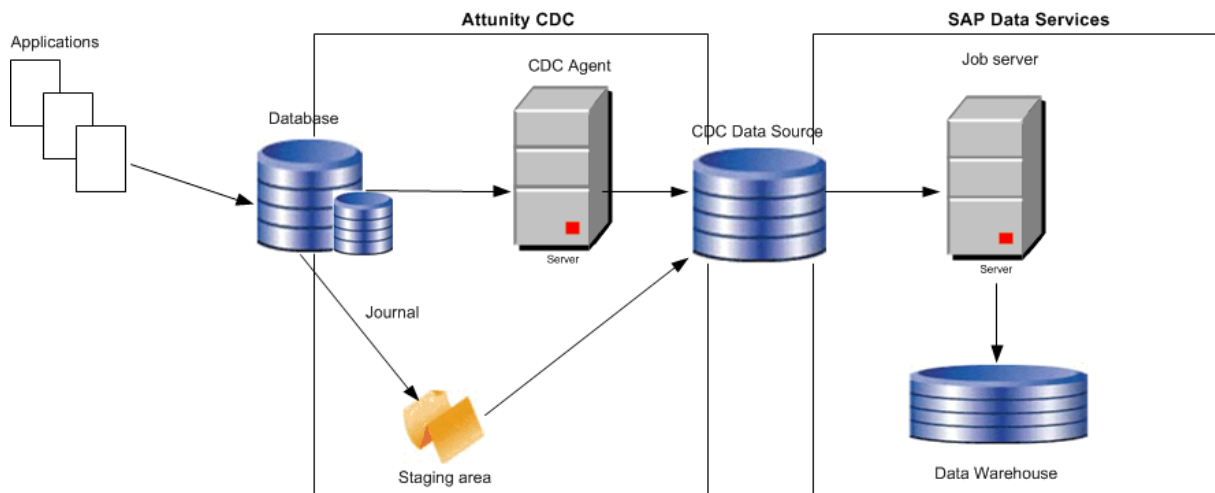
[Creating a data flow with an Oracle CDC source \[page 888\]](#)

29.5 Use CDC with Attunity mainframe sources

If your environment keeps large amounts of data current, the mainframe CDC feature is a simple solution to limiting the number of rows that must be read on a regular basis.

A source that reads only the most recent operations (INSERTS, UPDATES, DELETES) allows you to design smaller, faster delta loads.

SAP Data Services captures changed data on Attunity mainframe data sources and applies it to a target system. The following diagram shows the path that the data takes from Attunity CDC to SAP Data Services.



- The Attunity CDC Agent monitors the database journal for changes to specific tables. After the first request to capture changes, the CDC agent stores a context that the agent uses as a marker to not recapture changes prior to it.
- The CDC Agent sends the changed data to an optional staging area. The advantages of a staging area are:
 - A single journal scan can extract changes to more than one table. Without a staging area, multiple journal scans, one for each changed table, is required to extract changes.
 - Extracts only committed changes which is less processing than extracting every change. Less processing also occurs during recovery of a failed job because the recovery process does not need to back out the uncommitted changes.

However, a staging area requires additional storage and processing overhead. Refer to the Attunity CDC documentation for details.

- Attunity Connect CDC sends the changes to the CDC data sources through which the software can access the changes using standard ODBC or JDBC.

[Setting up Attunity CDC \[page 891\]](#)

[Setting up the software for CDC on mainframe sources \[page 892\]](#)

[Importing mainframe CDC data \[page 894\]](#)

Use a CDC datastore to import CDC table metadata.

[Configuring a mainframe CDC source \[page 895\]](#)

[Using mainframe check-points \[page 896\]](#)

[Limitations \[page 898\]](#)

29.5.1 Setting up Attunity CDC

If you currently use Attunity as the connection to SAP Data Services to extract data from mainframe sources, create an Attunity CDC data source in Attunity Studio. The following steps summarize the procedure for using the Attunity Studio wizard to create a CDC data source.

- Specify your data source.
- Based on your data source, choose one of the following methods to capture changes and specify the location of the journal:

- VSAM under CICS—By CICS Log stream
- DB2 on OS/390 and z/OS platforms—By DB2 Journal
- DB2 on OS/400—By DB400 Journal
- DISAM on Windows—By Journal

For a complete list of supported data sources, see the Attunity Connect CDC document.

- Select a name for your CDC agent.
- Specify if you want to capture before images for update operations. If you do not specify this option in Attunity Studio, you will not capture before images even if you specify the option [Get before-image for each update row](#).
- Select the tables to monitor for changes.

The Attunity Studio wizard generates the following components that you need to specify on the Datastore Editor when you define an Attunity CDC datastore:

- A CDC data source name that you specify in the option [Data source](#). Attunity generates the CDC data source on the same computer as the CDC agent by default. You have the option of placing the CDC data source on the client (same computer as SAP Data Services). Obtain the host name of this computer to specify in the option [Host location](#).
- A workspace for the CDC agent to manage the change capture event queue. You specify the workspace name in the option [Attunity workspace](#).

For more information, refer to the CDC setup section in the *Attunity Connect: The Change Data Capture Solution*.

Parent topic: [Use CDC with Attunity mainframe sources \[page 890\]](#)

Related Information

[Setting up the software for CDC on mainframe sources \[page 892\]](#)

[Importing mainframe CDC data \[page 894\]](#)

[Configuring a mainframe CDC source \[page 895\]](#)

[Using mainframe check-points \[page 896\]](#)

[Limitations \[page 898\]](#)

29.5.2 Setting up the software for CDC on mainframe sources

To use SAP Data Services to read and load changed data from mainframe sources using Attunity, do the following procedures on the Designer:

- Create a CDC datastore for Attunity
- Import metadata for Attunity tables
- Configure a mainframe CDC source
- Build real-time jobs using metadata

Parent topic: [Use CDC with Attunity mainframe sources \[page 890\]](#)

Related Information

[Setting up Attunity CDC \[page 891\]](#)

[Importing mainframe CDC data \[page 894\]](#)

[Configuring a mainframe CDC source \[page 895\]](#)

[Using mainframe check-points \[page 896\]](#)

[Limitations \[page 898\]](#)

29.5.2.1 Creating CDC datastores

The CDC datastore option is available for all mainframe interfaces to SAP Data Services.

Related Information

[Datastores \[page 92\]](#)

29.5.2.1.1 Creating a CDC datastore for Attunity

1. Open the Datastore Editor.
2. Enter a name for the datastore.
3. In the *Datastore type* box, select Database.
4. In the *Database type* box, select Attunity_Connector.
5. Check the *Enable CDC* box to enable the CDC feature. You can enable CDC for the following data sources. For the current list of data sources, refer to the Attunity web site.
 - VSAM under CICS
 - DB2 UDB for z/OS
 - DB2 UDB for OS/400
6. In the *Data source* box, specify the name of the Attunity CDC data source. You can specify more than one data source for one datastore, but you cannot join two CDC tables. You might want to specify multiple data sources in one Attunity datastore for easier management. If you can access all of the CDC tables through one Attunity data source, it is easier to create one datastore, enter the connection information once, and import the tables.

If you list multiple data source names for one Attunity Connector datastore, ensure that you meet the following requirements:

- Do not specify regular Attunity data sources with CDC data sources in the same datastore. The software imports data from regular Attunity data sources differently than from CDC data sources.
 - All Attunity data sources must be available by the same user name and password.
 - All Attunity data sources must use the same workspace. When you setup access to the data sources in Attunity Studio, use the same workspace name for each data source.
7. In the [Host location](#) box, specify the name of the host on which the Attunity data source daemon exists.
 8. In the [Port](#) box, specify the Attunity daemon port number. The default value is 2551.
 9. Specify the Attunity server workspace name that the CDC agent uses to manage the change capture event queue for the CDC data source.
 10. Complete the rest of the dialog and click [OK](#).

You can now use the new datastore connection to import metadata tables into the current repository.

Once saved, this datastore becomes a CDC datastore.

29.5.3 Importing mainframe CDC data

Use a CDC datastore to import CDC table metadata.

In the object library, right-click the datastore name and select [Open](#), [Import by Name](#), or [Search](#). For mainframe CDC, only the CDC tables that you selected when you set up an Attunity CDC are visible when you browse external metadata. Functions and templates are not available because the Attunity CDC datastore is read-only.

The SAP Data Services import operation adds the following columns to the original table:

Column name	Data type	Source of column
DI_SEQUENCE_NUMBER	integer	Generated by SAP Data Services
DI_OPERATION_TYPE	varchar(1)	Generated by SAP Data Services
Context	varchar(128)	Supplied by Attunity Streams
Timestamp	varchar(26)	Supplied by Attunity Streams
TransactionID	varchar(4)	Supplied by Attunity Streams
Operation	varchar(12)	Supplied by Attunity Streams
tableName	varchar(256)	Supplied by Attunity Streams

Task overview: [Use CDC with Attunity mainframe sources \[page 890\]](#)

Related Information

[Setting up Attunity CDC \[page 891\]](#)

[Setting up the software for CDC on mainframe sources \[page 892\]](#)

[Configuring a mainframe CDC source \[page 895\]](#)

[Using mainframe check-points \[page 896\]](#)

[Limitations \[page 898\]](#)

[Setting up Attunity CDC \[page 891\]](#)

29.5.3.1 The DI_SEQUENCE_NUMBER column

The `DI_SEQUENCE_NUMBER` column starts with zero at the beginning of each extraction. This field increments by one each time the software reads a row except when it encounters a pair of before- and after-images.

Both the before- and after-images receive the same sequence number. This sequencing column provides a way to collate image pairs if they become separated as a result of the data flow design.

You can configure Attunity Streams to retrieve before- images of UPDATE rows before the software applies the UPDATE operation to the target. Note that if you do not configure Attunity Streams to capture before- images in the database, the software will discard the rows.

If during the course of a data flow the before- and after-images become separated or get multiplied into many rows (for example, using GROUP BY or ORDER BY clauses in a query), you can lose row order.

The Map_CDC_Operation transform allows you to restore the original ordering of image pairs by using the `DI_SEQUENCE_NUMBER` column as its *Sequencing column*.

Related Information

[Using before-images \[page 887\]](#)

29.5.3.2 The DI_OPERATION_TYPE column

SAP Data Services generates values in the `DI_OPERATION_TYPE` column. Valid values for this column are:

- I for INSERT
- D for DELETE
- B for before-image of an UPDATE
- U for after-image of an UPDATE

29.5.4 Configuring a mainframe CDC source

When you drag a CDC datastore table into a data flow, it automatically becomes a source object.

Task overview: [Use CDC with Attunity mainframe sources \[page 890\]](#)

Related Information

[Setting up Attunity CDC \[page 891\]](#)

[Setting up the software for CDC on mainframe sources \[page 892\]](#)

[Importing mainframe CDC data \[page 894\]](#)

[Using mainframe check-points \[page 896\]](#)

[Limitations \[page 898\]](#)

29.5.4.1 Configuring a mainframe CDC table

1. Drag a CDC datastore table into a data flow.

The table automatically becomes a source object.

2. Click the name of this source object to open its Source Table Editor.
3. Click the CDC Options tab.
4. Specify a value for the *CDC subscription name*.

For more information, see the *Reference Guide*.

Related Information

[Using mainframe check-points \[page 896\]](#)

[Using before-images \[page 887\]](#)

29.5.5 Using mainframe check-points

Attunity CDC agents read mainframe sources and load changed data either into a staging area or directly into the CDC data source. Rows of changed data append to the previous load in the CDC data source.

When you enable check-points, a CDC job in SAP Data Services uses the subscription name to read the most recent set of appended rows and to mark the end of the read. If check-points are not enabled, the CDC job reads all the rows in the Attunity CDC data source and processing time increases.

To use check-points, on the Source Table Editor enter the *CDC Subscription name* and select the *Enable check-point* option.

If you enable check-points and you run your CDC job in recovery mode, the recovered job begins to review the CDC data source at the last check-point.

i Note

To avoid data corruption problems, do not reuse data flows that use CDC datastores because each time a source table extracts data it uses the same subscription name. This means that identical jobs, depending upon when they run, can get different results and leave check-points in different locations in the file. When you migrate CDC jobs from test to production, a best-practice scenario is to change the subscription name for the production job. Therefore, if the test job ever runs again, it does not affect the production job's results.

Task overview: [Use CDC with Attunity mainframe sources \[page 890\]](#)

Related Information

[Setting up Attunity CDC \[page 891\]](#)

[Setting up the software for CDC on mainframe sources \[page 892\]](#)

[Importing mainframe CDC data \[page 894\]](#)

[Configuring a mainframe CDC source \[page 895\]](#)

[Limitations \[page 898\]](#)

29.5.5.1 Using before-images from mainframe sources

The before image of an update row is the image of the row before the row is changed.

Follow these steps to capture before-image update rows:

1. Make sure Attunity Streams is set up to retrieve full before-images.
2. Select the [Get before-images for each update row](#) option in the CDC table's source editor.

The underlying, log-based CDC capture software must be set up properly, otherwise enabling the Get before-images for each update row option in the software has no effect.

After you check the [Get before-images for each update row](#) option, the software processes two rows for every update. In addition to the performance impact of this data volume increase, the before- and after-image pairs could be separated or lost depending on the design of your data flow, which would cause data integrity issues.

The Map_CDC_Operation transform can resolve problems, but undesirable results can still occur due to programming errors. When you use functions and transforms that re-order, re-direct, eliminate, and multiply the number of rows in a data flow, be aware of the possible impact to targets.

Related Information

[Using before-images \[page 887\]](#)

29.5.6 Limitations

The following limitations exist for this feature:

- You cannot use the following transforms and functions with a source table imported with a CDC datastore because of the existence of the SAP Data Services generated columns for CDC tables. The software cannot compare or search these columns.
 - Table_Comparison, Key_Generation, and SQL transforms
 - All database functions, such as `lookup`, `lookup_ext`, `key_generation`, `sql`, and `total_rows`
- You can only create one CDC source in a data flow.

Parent topic: [Use CDC with Attunity mainframe sources \[page 890\]](#)

Related Information

[Setting up Attunity CDC \[page 891\]](#)

[Setting up the software for CDC on mainframe sources \[page 892\]](#)

[Importing mainframe CDC data \[page 894\]](#)

[Configuring a mainframe CDC source \[page 895\]](#)

[Using mainframe check-points \[page 896\]](#)

29.6 Use CDC with SAP Replication Server

Data Services combined with SAP Replication Server's replication functionality provides two mechanisms to capture, transform, and propagate high volumes of data to a data warehouse in real time.

Using CDC with SAP Replication Server is ideal if your environment must keep large amounts of data current with minimal impact on operational systems. SAP Replication Server CDC sources in data flows read only the most recent operations (INSERTS, UPDATES, DELETES), allowing you to design smaller, faster delta loads.

The two methods for using SAP Replication Server are:

Method	Advantages
SAP PowerDesigner modeling	Allows you to run in batch mode in off-peak times to apply changes to the target database.

Method	Advantages
Built-in functions and a continuous work flow	You can take advantage of the following features with this method: • Auto configuration for provisioning real-time data capture. • Auto recovery in case of failure of job, using the auto-correction option in the CDC Reader. • Selective restore capabilities from Delta Queue of Replication Server. This can be helpful if the target database is ever corrupt and you need to bring the database to current state after restoring from Target database backup. • Automatic transition from initial load to delta-load. • Reader auto-correction functionality which enables the loader to load data to the target database with the bulk-load option.

[Overview for using a continuous work flow and functions \[page 899\]](#)

You can use the SAP Replication Server for changed-data capture using a continuous work flow and functions to capture data in real time. This method is a simpler alternative to using SAP PowerDesigner modeling.

[Overview for using the SAP PowerDesigner modeling method \[page 904\]](#)

You can use the SAP Replication Server for changed-data capture using SAP PowerDesigner modeling. This method is an alternative to using built-in functions with a continuous work flow.

29.6.1 Overview for using a continuous work flow and functions

You can use the SAP Replication Server for changed-data capture using a continuous work flow and functions to capture data in real time. This method is a simpler alternative to using SAP PowerDesigner modeling.

The process is as follows:

- Data Services connects directly to the source database using its native connector (for example the OCI library for Oracle) to:
 - Import source table metadata into its repository
 - Initially load the target database
- Data Services communicates with the Replication Agent using ct_lib to:
 - Validate required global configurations
 - Mark source tables for replication
 - Obtain Replication Server data-type mapping to the source database
- Data Services communicates with the Replication Server using ct_lib to:
 - Configure the target queue
 - Configure Replication definitions and subscriptions for the source database
 - Reads CDC data from the queue

The process for enabling the environment is as follows:

1. Ensure the prerequisite software is installed.
2. Configure the Replication Server and Agent to work with Data Services.
3. In Data Services, configure the datastore connections and import the tables to be monitored.

4. In Data Services, design the CDC job using a continuous work flow and provided built-in functions.

Parent topic: [Use CDC with SAP Replication Server \[page 898\]](#)

Related Information

[Overview for using the SAP PowerDesigner modeling method \[page 904\]](#)

29.6.1.1 Prerequisites

To enable changed-data capture using the SAP Replication Server and Data Services built-in functions with a continuous work flow, ensure you have installed the following prerequisites.

Component	Function	Reference
SAP Replication Server	Captures changed data from the source database and stores it in queues on the Replication Server, which can then be retrieved by a Data Services job.	SAP Sybase online documentation at http://infocenter.sybase.com/help/index.jsp 
Replication Agent	Communicates between the source database and the Replication Server.	Refer to the <i>Replication Agent Primary Database Guide</i> for your source database (SAP Sybase online documentation)
Data Services Designer	Use to design the continuous work flow to capture changed data.	

29.6.1.2 Configuring the Replication Server and Agent

Observe the following requirements when using the Replication Server and Replication Agent for CDC.

In general, it is recommended you use a dedicated Replication Server/Heterogeneous Replication Agent instance for capturing changed data because Data Services jobs manage their own configurations. In order to auto-configure, jobs suspend/resume the Heterogeneous Replication Agent and operation retention connections.

Replication Server

- Set the Replication Server character set to UTF8.
- Data Services must be able to connect directly to the primary Replication Server to manage configurations of the operation retention connection queue, replication definition and subscription, and consume changed data from the queue.

- The Data Services user that logs in to the Replication Server must have the following permissions:
 - connect target
 - create object
 - primary subscribe
- It is recommended that the Replication Server log file not be in the same file system as the partition file because large volumes of replicated data in the partition could interfere with log file size. You can change the location of the log file in one of the following places:
 - rs_init: edit the rs.rs_rs_errorlog parameter. For example: `rs.rs_rs_errorlog: /home/user/sample_repserver/sample_rs.log`
 - If you use the RUN file to start the Replication Server, edit the RUN file to change the log location. For example: `-E/home/user/sample_repserver/sample_rs.log`
- On the Replication Server, you can configure a partition that expands automatically. For example:

```
create auto partition path partition2 on '/home/.../partition2' with auto
expand size = 1024 max size = 10240
```

The logical partition, partition2, is created in a directory called “partition2”. If necessary, a partition file with 1024 Mb is created under partition2 until the limit is reached. To check for an auto partition on the Replication Server:

```
admin auto_part_path
```

For details on the usage of this feature, refer to the *Replication Server Reference Manual*.

- Refer to the latest Replication Server sizing guide for general sizing guidelines.

Replication Agent

- Data Services requires the Administrative user and password to connect to the heterogeneous Replication Agent.
- The SAP ASE Replication Agent user must have the rs_replication role.

Note

Data Services changes global configuration and table level configurations that are required to capture and process change data.

Troubleshooting

If replication does not happen as expected, check the following:

- Check the Replication Agent log to determine if and why the agent might not be running.
- The partition on the Replication Server is full. Check the Replication Server log for a message such as:

```
SQM_ADD_SEGMENT ('102:1'): Going to wait for a segment to be freed.
Administrative action is needed
```

To resolve this issue, execute the following command on the Replication Server:

```
alter partition <partition name> expand size = <number of segments>
```

To check the current disk space:

```
admin disk_space
```

- Memory is low on the Replication Server. Check the Replication Server log for a message such as:

```
Replication Agent for rao21.orcl1 is sleeping due to memory controls (EXEC threshold '90 percent') being triggered.
```

To resolve this issue, execute the following command on the Replication Server:

```
configure replication server set memory_limit to '<N megabytes>' :
```

To check the current memory usage:

```
admin stats, mem_in_use
```

- If you are using the embedded replication server system database (ERSSD) as a source and discover it is consuming a large amount of physical memory, you can limit the cache size using the SQL Anywhere `-ch` switch as follows:
 1. Shut down the Replication Server instance.
 2. Open the Replication Server instance `<rs_instance_name>.cfg` file (for example, `SAMPLE_RS.cfg`), and add the `erssd_start_cmd` parameter to specify the command to start ERSSD (the `erssd_start_cmd` parameter lets you specify a different command to start ERSSD). The following example shows how to modify the SQL Anywhere ERSSD default configuration to limit the cache size to 1 GB for the existing ERSSD database.

```
erssd_start_cmd=$SYBASE/$SYBASE_REP/ASA16/bin/dbspawn -f -q dbsrv16 -s  
none -ti 0 -x  
"tcpip(PORT=11751;DOBROAD=NO;BLISTENER=NO)" -ch 1g -o $SYBASE/$SYBASE_REP/  
/samp_repserver/errorlog/SAMPLE_RS_ERSSD.out $SYBASE/$SYBASE_REP/  
samp_repserver/dbfile/SAMPLE_RS_ERSSD.db
```

3. Restart the Replication Server instance.

For details about the `erssd_start_cmd` parameter, refer to the *Replication Server Configuration Guide*.

29.6.1.3 Creating the CDC datastore

This procedure describes how to create a datastore for using the Replication Server to capture changed data from supported source (primary) databases.

Add a source datastore connection to the source (primary) database and import the desired source tables.

1. From the [Datastores](#) tab of the object library, open the datastore editor (right-click and select [New](#)).
2. Enter a name for the datastore.
3. For [Datastore type](#), select [Database](#).
4. For [Database type](#), select the source database.
5. In the drop-down box next to the database type, select [Replication Server CDC](#).
6. Select the [Database version](#).
7. Enter a [Database server name](#).

8. Enter a database *User name* and *Password*.
9. Click *Advanced* and enter the following CDC values:

Option	Description
Replication Server name	The Replication Server name must match what is defined in the interface file <code>sql.ini</code> . In addition, the name must match the name of the Replication Server to which the Replication Agent is connected.
Replication Server user name	The Replication Server user must have "create object", "primary subscribe" and "connect target" permission rights.
Replication Server password	The password to access the system.
Retention Period	Set the value that corresponds to the Replication Server connection's <code>save_interval</code> property in minutes. For example, if your target database back-up interval is 24 hours, set the Retention Period to $24 \times 60 = 1440$ to set the equivalent number of minutes to retain the last 24 hours of changed data.
Replication Agent name	The Replication Agent name must match what is defined in the interface file <code>sql.ini</code> .
Replication Agent user name	The Replication Agent user must be an 'sa' user. If using the SAP ASE source database, the Replication Agent user must be assigned to 'replication_role'.
Replication Agent password	The password to access the system.

10. Click *Apply* or *OK* to save the datastore.

29.6.1.4 Building the data flow

This procedure describes how to build a data flow for capturing changed data using built-in functions and a continuous work flow.

- Add a source datastore connection to the source (primary) database and import the desired source tables.
 - Add a CDC datastore and import the corresponding table(s).
1. Add a new job.
 2. Add a global variable called, for example, `$initialload` with a default value of 0.
 3. Add a conditional (for example called `initial_load`) with the following parameters:
 - a. If `$initialload = 1`
 - b. Add a script called, for example, `initialloadCCDmarker`.
 - c. In the script, add a call to the pre-defined function `begin_initial_load()`.
 - d. Add a data flow called, for example, `initial_load`.
 - e. Open the data flow and add the original source table as the source.
 - f. Complete the data flow by adding a target (additional transforms are optional).
 - g. Open the loader options and select *Delete data from table before loading*.
 4. Return to the job and add a continuous work flow.
 5. Open the continuous work flow:

- a. Add a script called, for example, MarkBeginCDCLoad.
 - b. In the script, add a call to the pre-defined function begin_delta_load().
6. Add a CDC data flow:
 - a. Add a CDC table from the CDC datastore that corresponds to the source table used in the initial_load data flow.
 In the source editor, optionally select *Enable auto-correct*. This option can be useful when the loader can identify source table records in the target database using keys in the source table to apply auto-correction operations that the reader generates.
 - a. Add a Map_CDC_Operation transform.
 - b. Add a target and connect the objects.
7. When you execute the job for the first time, on the *Global Variable* tab set the \$initialload variable to 1 to initialize (or reinitialize) the target datastore.

Changed data replicates to the target.

If the target database gets corrupted because of an error, you can restore the changed records from the Replication Server backlogged transactions. Use the Data Services built-in function restore_repserver_cdb_backlogged_transactions.

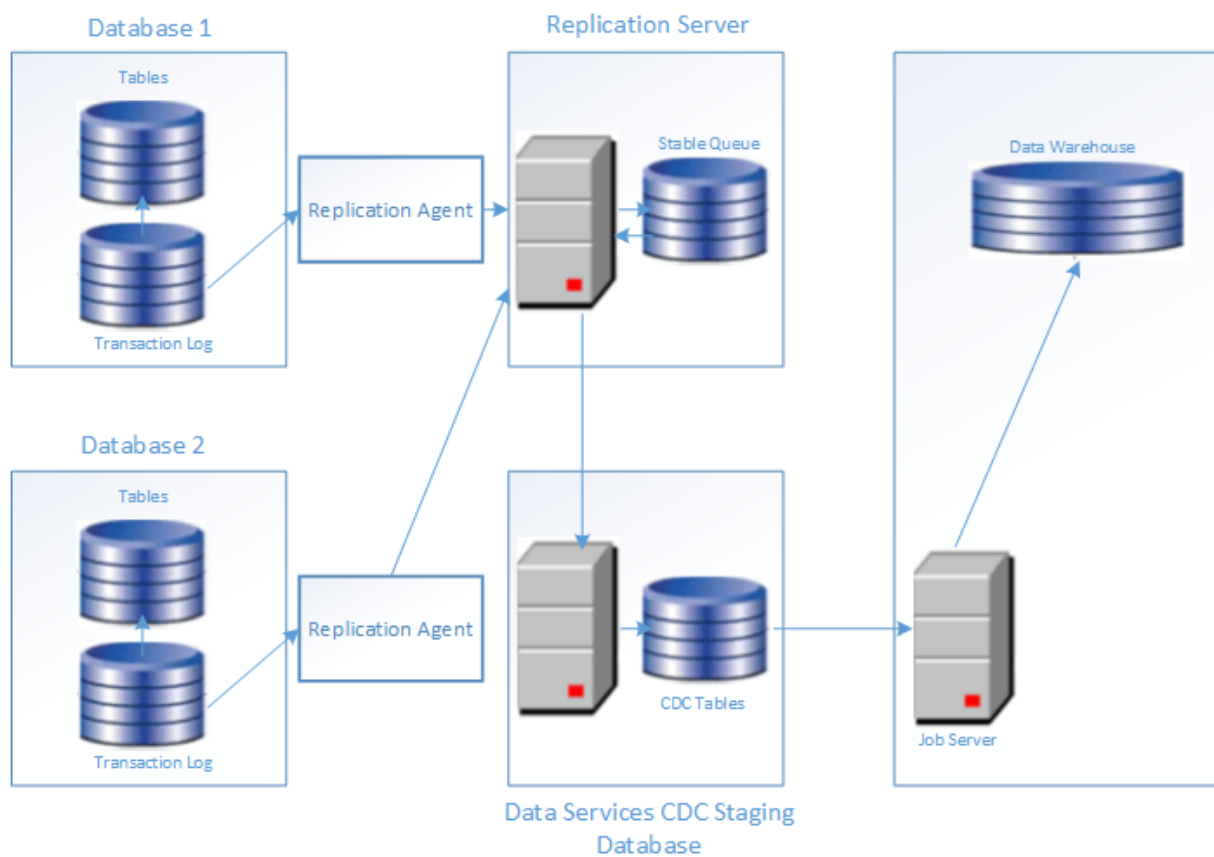
Related Information

[Reference Guide: restore_repserver_cdb_backlogged_transactions](#)

29.6.2 Overview for using the SAP PowerDesigner modeling method

You can use the SAP Replication Server for changed-data capture using SAP PowerDesigner modeling. This method is an alternative to using built-in functions with a continuous work flow.

Using the SAP PowerDesigner Data Movement Model, Data Services manages the CDC environment by building a CDC staging database and creating SAP Replication Server replication definitions and subscriptions. Upon deployment of the replication definitions and subscriptions and the CDC staging database definitions, SAP Replication Server publishes changed data from the source table to its corresponding CDC table that resides in the CDC staging database. The CDC table source in the Data Services data flow reads the most recently changed data from the last check-point from the CDC table and applies it to the target. The following diagram shows how the changed data moves from the Replication Server to Data Services.



Parent topic: [Use CDC with SAP Replication Server \[page 898\]](#)

Related Information

[Overview for using a continuous work flow and functions \[page 899\]](#)

29.6.2.1 Prerequisites

To enable changed-data capture using the SAP Replication Server and SAP PowerDesigner modeling, ensure you have installed the following components.

Design time component	Function	Reference
SAP PowerDesigner	Modeling tool for creating the replication definitions, the subscriptions for SAP Replication Server, and the Data Services CDC staging database definitions. You need this tool along with Data Services Designer to model the source system to the target system CDC data flow.	SAP Sybase online documentation at http://infocenter.sybase.com/help/index.jsp 
Data Services Designer	Use to design the data flow for transforming changed data before applying to the target system.	

Run-time component	Function	Reference
SAP Replication Server	Captures changed data from the source database and publishes to the Data Services CDC staging database.	SAP Sybase online documentation
isql tool	Deploys replication definitions and subscriptions in the Replication Server, enabling the source tables in the source databases for replication, and creates the Data Services CDC database definitions in staging database.	<i>Utility Guide of Adaptive Server Enterprise</i>
SAP ASE	Stages Data Services CDC changed data.	SAP Sybase online documentation
SAP ASE Job Scheduler	Purges the old changed data from the CDC staging database (part of CDC environment management).	i Note Install the Job Scheduler Agent as part of SAP ASE 15.7.1.
Data Services Job Server	Transforms the changed data and loads it to the target.	

29.6.2.2 Modeling of Replication Process for Data Services CDC Staging Database

Power Designer automates the process of configuring SAP Sybase Replication Server, which includes marking the source table as replicated, creating the replication definitions and subscriptions and defining the Data Services CDC staging database.

i Note

Tables in source database that needs to be replicated, must have primary key.

The following steps summarize the procedure to model the replication process for Data Services CDC Staging database in Power Designer:

1. Drop the `SybaseRepServerDataServicesCDC100.xem` file from Data Services install location
`%<DS_COMMON_DIR>%/ext/cdc` into Power Designer's ► [Resource Files](#) ► [Extended Model Definitions](#) ► folder.

i Note

Data Services Power Designer Extension is a part of the Data Services Package.

2. Create a new Data Movement Model in the Power Designer by selecting ► [Model](#) ► [New](#) ► [Data Movement Model](#) ►. Enter the *Model name* and in *Extensions* choose [Replication Server 15.5](#) from the drop down list.
3. Attach [SAP Data Services CDC 14.2.0 extension](#) to it. To do this, go to ► [model](#) ► [extensions](#) ► [List of Extensions](#) ►. Click the attach an extension icon and choose [SAP Data Services CDC 14.2.0 extension](#).

i Note

This extension works only with Replication Server 15.7 data movement modeling.

4. Run the [Replication Wizard](#) in the Power Designer by selecting ► [Tools](#) ► [Replication Wizard](#) ►. It creates the replication server process, the source physical data model and the replicated physical data model.

i Note

The Replication Wizard reverse-engineers the source databases to import database objects into the data movement model. Choose only tables and following table properties: Primary Keys, Foreign Keys, Alternate Keys and indexes to create the replicated physical database model.

5. When you are on the [Publication Selection](#) window, in the [Replication Wizard](#) dialog box, select [Create or select a single publication to gather replicated tables](#) and choose replication type as [replication definitions](#).

i Note

When the Replication Wizard finishes, Power Designer creates default replication definitions (called Articles) and the Physical Data Model for the Remote database with all selected tables from the source Physical Data Model and corresponding table columns and properties.

6. In the [Replication Wizard](#), when you are on the [Remote Physical Data Model Selection](#) screen, choose the [DBMS](#) as [Sybase AS Enterprise 15.7](#).
7. Save the Data Movement Model and Physical Data Model.
8. Select the Replication Process icon in the Data Movement Model diagram and choose [Add Data Services CDC Definitions](#) from context menu to run Data Services extension to create the following:
 - a. CDC replication definitions
 - b. CDC staging tables
 - c. CDC system tables (DS_CDC_TABLES_MAP, DS_CHANGE_RETENTION and DS_PURGE_TIMESTAMP)
 - d. User SAPDS

i Note

Selecting the [Add Data Services CDC Definitions](#) option automatically changes owner of the replicated tables, the purge stored procedure, the CDC staging table and the CDC system tables to SAPDS. It

defines the CDC staging table name same as the Remote Table name of the article with suffix "_SAPCDC". It creates or overrides function strings: rs_delete, rs_insert, rs_update and rs_get_textptr for blob columns. The CDC staging table defines additional columns in addition to the RemoteTable columns to store the Replication Server system variables:

Function string	What it does
DS_SRCDB	stores system variable "rs_origin_db"
DS_SRCDB_SRV	stores system variable "rs_origin_ds"
DS_SRCDB_ID	stores system variable "rs_origin"
DS_SRCDB_COMMIT_TIME	stores system variable "rs_origin_commit_time"
DS_SRCDB_ORIGIN_QID	stores "rs_origin_qid"
DS_SRCDB_OP_TYPE	stores change operation types "I"- insert , "B"- Before image of update, "U"- After image of update, "D"-Delete
DS_CDCDB_PUBLISH_TIME	stores publish datetime of CDC database when record get inserted by the Replication Server

[Add Data Services CDC Definitions](#) command also sets "CDC Staging Table" attributes with the CDC staging table name for each article.

Additionally, [SAP Data Services CDC 14.2.0](#) extension adds two extended attributes, "Retention Period" and "CDC Staging Table" to Article object. Default value "Retention Period" extended attributes is 14 days. You can change this value.

29.6.2.3 Configuring the environment

The following process describes how to configure Data Services and Replication Server to capture changed data using the SAP PowerDesigner modeling method.

1. Create the Data Services CDC staging database in the SAP ASE server.
2. Configure the Replication Server to add the source databases and the CDC staging database.
3. Configure the Data Services CDC staging database runtime environment in the SAP ASE server.
4. Deploy the replication definitions and subscriptions into the Replication Server.
5. Configure the primary databases runtime environment.

29.6.2.3.1 Creating the Data Services CDC staging database in SAP ASE server

The following procedure describes how to create a Data Services CDC staging database in SAP ASE server.

1. Log in as database administrator and create a database that matches the name of Data Services CDC staging database as defined in Power Designer's data movement model.

2. Create SAPDS log-in user. Lock this user to prevent log-in.

i Note

This user is reserved for Data Services CDC and should not be used for any other purpose. All the Data Services staging database objects are owned by user SAPDS.

3. Select the Data Services staging database Physical Data Model in Power Designer's Data Movement Model, and right-click the context to select [Generate Database](#) to generate the CDC database DDL script.

i Note

This automatically generates DDLs of all the database objects that are required for the Data Services CDC staging database.

4. Check the CDC database DDL script that is generated as a file.
5. Deploy the DDL script using the isql tool. Enter the code:

```
isql-Usa-Ppwd -Spds -Dpdb -i DS_CDC_staging_DDL_script.sql-  
oDS_CDC_staging_DDL_script.out
```

where:

- **<sa>** is the system administrator user login on the Data Services CDC staging data server.
- **<pwd>** is the password for the system administrator user login.
- **<pds>** is the name of the Data Services CDC staging data server.
- **<pdb>** is the name of the Data Services CDC staging database.

29.6.2.3.2 Configuring Replication Server to add source databases and CDC staging database

The following procedure describes how to configure Replication Server to add a source database and CDC staging database.

1. Add the source database and the CDC staging database to the interface file in Replication Server host.
2. Run `rs_init` to add the source database and the CDC staging database.

29.6.2.3.3 Configuring Data Services CDC staging database in SAP ASE server

The following procedure describes how to configure the Data Services CDC staging database in the SAP ASE server.

1. Select the Data Services staging database's Physical Data Model in PowerDesigner's Data Movement Model, and right-click the context to select [Generate Script](#).
2. Select the [SAP Data Services CDC 14.2.0](#) extension in the dialog box. It generates a script file suffixed with `_DataServices_CDC_DB_Runtime_Script.sql`. This script file creates the Data Services CDC staging database runtime environment and provides the following functionalities:

- a. Validates the staging database to verify if it has been added to the Replication Server by its rs_init utility.
 - b. Validates if the Data Services CDC staging database create script has run.
 - c. Creates the Replication Server maintenance user that was defined in the Data Movement Model (if it was not created before). You can change the password after creation to protect unauthorized access.
 - d. Adds the maintenance user to database.
 - e. Adds the TargetTable name and the CDC staging table name to DS_CDC_TABLES_MAP table for each replication definition.
 - f. Adds the CDC staging table name and the retention period to DS_CHANGE_RETENTION table for each replication definition.
 - g. Grants select, insert, update, and delete privileges to the CDC staging tables.
 - h. Grants execute privilege to the PURGE_OLD_CHANGE_DATA stored procedure.
 - i. Grants the Replication Server maintenance user the privileges to execute the PURGE_OLD_CHANGE_DATA scheduled job in the SAP ASE Job Scheduler database.
 - j. Creates PURGE_OLD_CHANGE_DATA scheduled job in SAP ASE Job Scheduler database. The job name is SAPDS_<CDC Staging Database Name>_PURGE_CHANGE_OLD_DATA.
3. Deploy the DDL script using the isql tool.

Database user who logs in must be a super user. Enter the code:

```
isql -Usa -Ppwd -Spds -Dpdb -i DS_CDC_staging_Runtime script.sql -o
DS_CDC_staging_Runtime_script.out
```

where:

- <sa> is the system administrator user login on the Data Services CDC staging data server.
- <pwd> is the password for the system administrator user login.
- <pds> is the name of the Data Services CDC staging data server.
- <pdb> is the name of the Data Services CDC staging database.

29.6.2.3.4 Deploying replication definitions and subscriptions into the Replication Server

The following procedure describes how to deploy replication definitions and subscriptions into the Replication Server.

1. Select the Replication Process in the PowerDesigner Data Movement Model, and right-click the context to select *Generate Script*.
2. Set the *Create Connection* value to *false* in the dialog box, and click *OK* to generate the replication definitions, CDC function string definition subscriptions, and the Replication Server RCL script file.

i Note

It is recommended that you create a database connection using the rs_init tool.

3. Deploy the RCL script using the isql tool into the Replication Server that you configured for the CDC staging database and the source databases.

The Replication Server user who logs in to deploy must be a super user. Enter the code:

```
isql -Usa -Ppwd -Spds -Dpbd -i RCL_script.sql -o RCL_script.out
```

where:

- **<sa>** is the system administrator user login on the Data Services CDC staging data server.
- **<pwd>** is the password for the system administrator user login.
- **<pds>** is the name of the Data Services CDC staging data server.
- **<pdb>** is the name of the Data Services CDC staging database.

29.6.2.3.5 Configuring the primary database environment

The following procedure describes how to configure the primary database environment.

1. Select the primary database in the PowerDesigner Data Movement Model, right-click the context to select [Generate Script](#), and select [Replication Server 15.7](#). It generates three script files.
2. Select the script file that includes the sp_setreptable stored procedure command.
3. Check the script to make sure that the owner_on parameter is set for articles whose primary table name is qualified by owner.

The SAP Data Services CDC 14.2.0 extension turns on the multiple-owner attribute for all primary tables that are qualified by owner.

4. Deploy these scripts using the isql tool and the appropriate source database tool to create user, grants and permissions.

The database user who deploys the script must be a super user. Enter the code:

```
isql -Usa -Ppwd -Spdb -Dpdb -i Primary_database_Runtime script.sql -o  
Primary_database_Runtime script.out
```

where:

- **<sa>** is the system administrator user login on the Data Services CDC staging data server.
- **<pwd>** is the password for the system administrator user login.
- **<pds>** is the name of the Data Services CDC staging data server.
- **<pdb>** is the name of the Data Services CDC staging database.

This marks primary tables as replicable, and changed data can be captured by the Replication Agent and propagated to the Replication Server.

29.6.2.4 Configuring Data Services

To use Data Services to read and load changed data from SAP Sybase Replication Server using the PowerDesigner modeling, do the following procedures in the Designer.

- Create a CDC datastore.
- Import the metadata for SAP Replication Server tables.
- Create and define a data flow.

- Configure a CDC source.

29.6.2.4.1 Creating a CDC datastore and importing table metadata

This procedure describes how to create a CDC datastore using the SAP Replication Server. To access the source, create a CDC datastore using the Designer. A CDC datastore is a read-only datastore that can only access tables. After you create the CDC datastore, you use it to import CDC table metadata.

1. From the [Datastores](#) tab of the object library, open the datastore editor.
2. Enter a name for the datastore.
3. For [Datastore type](#), select [RepServer CDC](#).
4. For [Database type](#), select [SAP ASE](#).
5. Select the [Database version](#).
6. Enter a [Database server name](#).
7. Enter a [Database name](#).
8. Enter a database [User name](#) and [Password](#).
9. To create more than one configuration for this datastore, click [Apply](#), then click [Edit](#) and enter the configuration details.

i Note

You cannot change the database type, version, or CDC method.

10. Click [OK](#).
11. To import the tables, in the object library, right-click the datastore name and select [Open](#), [Import by Name](#), or [Search](#).
Only the CDC tables that you selected when you set up SAP Replication Server for CDC are visible when you browse external metadata. You must create a CDC table in SAP ASE for every table you want to read before you can browse and import that CDC table using SAP Data Services.

Related Information

[Defining a database datastore \[page 96\]](#)

29.6.2.4.2 Creating and defining the data flow

This procedure describes how to create a data flow to read changed data using SAP Replication Server and the SAP PowerDesigner modeling.

1. In the object library, drag the CDC table, a Query transform, and a Map_CDC_Operation transform to the workspace.

i Note

- Multiple Replication Server CDC tables are allowed in same data flow.
- The Replication Server CDC table stores data records extracted from the source database log. However, these stored data records are not equivalent to the records available in the corresponding source table. Additionally, captured changed data records older than the retention period are also deleted from the Replication Server CDC table. Therefore, joining two or more such CDC tables may not produce the same result as joining their corresponding source database tables.

2. Configure the CDC table as described in the next section.
3. Add the appropriate target table and connect the objects.
4. Open the Query Editor.
5. Map the Data Services control columns and the source table columns that you want in your target.

The Map_CDC_Operation transform uses the values in the column in the *Row operation column* box to perform the appropriate operation on the source row for the target table. For the CDC source table, the DI_OPERATION_TYPE column is automatically selected as the *Row operation column*.

The operations can be INSERT, DELETE, or UPDATE. For example, if the operation is DELETE, the corresponding row is deleted from the target table.

Related Information

[Reference Guide: Map_CDC_Operation](#)

29.6.2.4.3 Configuring CDC source table

This procedure describes how to configure a CDC source table. Data Services can apply check-points across all the tables (in a given datastore configuration) for all the data flows in a job to provide data consistency.

1. Drag a CDC datastore table into a data flow.
2. Click the name of this source object to open its editor.
3. Click the *CDC options* tab.
4. Optionally select *Enable check-point*.

Once a check-point is placed, the next time the CDC job runs, it reads only the rows inserted into the CDC table since the last check-point.

If check-points are not enabled, the CDC job reads all the rows in the CDC data source (with increased processing time).

Related Information

[Reference Guide: CDC table source](#)

29.7 Use CDC with Microsoft SQL Server databases

SAP Data Services utilizes three methods of change data capture in Microsoft SQL Server databases.

SAP Data Services captures changed data on Microsoft SQL Server databases and apply it to target systems using the following methods:

- Changed-Data Capture
- Change Tracking
- Replication Server

All three methods use the concept of check-points to read the most recent set of changes and mark the end of the read. If check-points are not enabled, Data Services reads all the changes. In addition, if you enable check-points and you run your CDC job in recovery mode, the recovered job begins to review the CDC data source at the last check-point.

To capture changed data:

- Enable CDC in your Microsoft SQL Server database
- Enable CDC on the source tables
- Configure CDC in Data Services datastores, jobs, and sources

The following table compares the three CDC methods. Refer to the Microsoft SQL Server documentation for version information and for complete information about the three methods.

Feature	Changed-Data Capture	Change Tracking	Replication Server
Microsoft SQL Server supported version	2008 and later	2008 and later	2000, 2005, 2008
Synchronous (immediate; asynchronous has latency)	No The SQL Server CDC capture job captures changes.	Yes Data Services tracks changes in line with INSERT, UPDATE, and DELETE operations so changes are available immediately.	No Changes are available after the the log reader agent transfers the operations into the distribution database.
Requires SQL agent	Yes A capture job populates the CDC tables.	No The database engine populates the primary keys into change tables during DML operations.	Yes A replication log reader agent handles the replication.
Automatic cleanup process	Yes, periodically	Yes, periodically	Yes, periodically

Feature	Changed-Data Capture	Change Tracking	Replication Server
Requires separate tables to store tracking data	Yes Tracking data stored in a capture table. The storage depends on the number of columns captured for CDC.	Yes Uses one internal change tracking table per source table. Uses one transaction table per database. The storage depends on number of primary key columns of the source table.	Yes Stored in a distribution database.
Historical data available	Yes	No	Yes
Requires primary key	No	Yes	Yes
Before image available for UPDATE operation in Data Services	Yes Data Services automatically reads the before-image and makes it available for further use in the data flow.	No	Yes (optional)
Recommendation	Use when historical data or consistency across multiple tables is required.	Use for rapid synchronization.	Supported but deprecated in newer versions.

[Limitations \[page 916\]](#)

SAP Data Services has specific limits for using the change data capture methods in Microsoft SQL Server.

[Data Services columns \[page 917\]](#)

[Changed Data Capture \(CDC\) method \[page 919\]](#)

The Changed Data Capture (CDC) method applies check points to all tables that you use in a job to provide data consistency.

[Change Tracking method \[page 923\]](#)

The Change Tracking method identifies that rows in a table have changed but ignores how many times the row has changed or the values of any intermediate changes.

[Replication Server method \[page 926\]](#)

The Replication Server method stores changes in a distribution database stored in the Replication Server.

29.7.1 Limitations

SAP Data Services has specific limits for using the change data capture methods in Microsoft SQL Server.

Limits on transforms and functions

Source tables imported with a CDC datastore have Data Services-generated columns for CDC tables, which Data Services cannot compare or search. Therefore, you cannot use the following transforms and functions with these source tables.

Transforms:

- Table_Comparison
- Key_Generation
- SQL
- Data_Transfer

i Note

The History_Preserving transform is not supported with the Change Tracking method.

All database functions including the following:

- lookup
- lookup_ext
- lookup_seq
- search_replace
- pushdown_sql
- truncate_table
- key_generation
- sql
- total_rows

Limitations with source tables

The following are additional limitations in relation to the source tables imported with a CDC datastore:

- You can create only one CDC source table in a data flow.
- You cannot use CDC tables in real-time jobs.
- You cannot use profiling for CDC tables.
- You cannot display optimized SQL for data flows that read CDC tables because optimized SQL is not available.
- You cannot export or import jobs with CDC tables because those jobs do not retain the check-point information saved in the Data Services repository.

Additionally, for the Change Tracking method, because Microsoft SQL Server cannot track the TRUNCATE operation on a source table, Data Services cannot propagate those results to the target.

Using the same subscription name

For the Change Tracking and Replication Server methods, do not use the same subscription name to read the same CDC table in parallel data flows. Data Services identifies the check-point by a combination of the subscription name and table name. If you reuse data flows that use CDC datastores, each time a source table extracts data, it uses the same subscription name. This means that identical jobs, depending upon when they run, can get different results and leave check-points in different locations in the file.

Parent topic: [Use CDC with Microsoft SQL Server databases \[page 914\]](#)

Related Information

[Data Services columns \[page 917\]](#)

[Changed Data Capture \(CDC\) method \[page 919\]](#)

[Change Tracking method \[page 923\]](#)

[Replication Server method \[page 926\]](#)

29.7.2 Data Services columns

When you import metadata into a CDC datastore using any of the methods, Data Services adds the following columns to the original table:

- DI_SEQUENCE_NUMBER
- DI_OPERATION_TYPE

[DI_SEQUENCE_NUMBER column \[page 918\]](#)

If image pairs become separated as a result of the data flow design, the DI_SEQUENCE_NUMBER column provides a way to collate image pairs.

[DI_OPERATION_TYPE column \[page 918\]](#)

SAP Data Services generates values in the DI_OPERATION_TYPE column to indicate the type of operation performed on the record.

Parent topic: [Use CDC with Microsoft SQL Server databases \[page 914\]](#)

Related Information

[Limitations \[page 916\]](#)

[Changed Data Capture \(CDC\) method \[page 919\]](#)

[Change Tracking method \[page 923\]](#)

[Replication Server method \[page 926\]](#)

29.7.2.1 DI_SEQUENCE_NUMBER column

If image pairs become separated as a result of the data flow design, the DI_SEQUENCE_NUMBER column provides a way to collate image pairs.

The DI_SEQUENCE_NUMBER column starts with zero at the beginning of each extraction. The column increments by one each time the software reads a row except when it encounters a pair of before- and after-images. Both the before- and after-images receive the same sequence number. If during the course of a data flow the before- and after-images become separated or get multiplied into many rows, you could lose row order. For example, when you use GROUP BY or ORDER BY clauses in a query, the before- and after-images may become separated.

The Map_CDC_Operation transform enables you to restore the original ordering of image pairs by using the DI_SEQUENCE_NUMBER column as its *Sequencing column*.

For the Replication Server method, configure the server to retrieve before-images of UPDATE rows before Data Services applies the UPDATE operation to the target. Note that if you do not configure the Replication Server to capture before-images in the database, Data Services captures only after-images by default. For information about when to consider using before-images, see [Using before-images \[page 887\]](#).

Parent topic: [Data Services columns \[page 917\]](#)

Related Information

[DI_OPERATION_TYPE column \[page 918\]](#)

[Map_CDC_Operation](#)

29.7.2.2 DI_OPERATION_TYPE column

SAP Data Services generates values in the DI_OPERATION_TYPE column to indicate the type of operation performed on the record.

Data Services generates the following values in the DI_OPERATION_TYPE column:

- I for INSERT
- D for DELETE
- B for before-image of an UPDATE (except for the Change Tracking method, which does not use a before-image)
- U for after-image of an UPDATE

Parent topic: [Data Services columns \[page 917\]](#)

Related Information

[DI_SEQUENCE_NUMBER column \[page 918\]](#)

29.7.3 Changed Data Capture (CDC) method

The Changed Data Capture (CDC) method applies check points to all tables that you use in a job to provide data consistency.

When you use the CDC method, Data services adds check points across all tables from a specific database datastore that you use in the data flow or data flows in a job.

The process to use CDC includes creating a datastore strictly for the purpose of importing CDC metadata from Microsoft SQL Server. After you import the CDC metadata, you configure a source table using CDC, then you use the CDC tables in a WHILE loop in a data flow.

Enable CDC first in the datastore editor, then select the *Enable check-point* option in the data flow source table editor. If *Enable check-point* is not selected, Data Services retrieves all the available data at that time for that table.

This CDC method is not available with Microsoft SQL Server versions before 2008.

[Creating a CDC datastore \[page 920\]](#)

Before you enable the Changed Data Capture (CDC) method, create a datastore connection to Microsoft SQL Server to import the metadata of a capture instance into the current repository.

[Importing CDC metadata \[page 921\]](#)

Use the Changed Data Capture (CDC) datastore that you created for the capture instance to import metadata.

[Configuring a source table using CDC \[page 921\]](#)

A source table is required when you use the Changed Data Capture (CDC) method for Microsoft SQL Server data.

[Using CDC for data flows in a WHILE loop \[page 922\]](#)

If a data flow runs in a WHILE loop to retrieve changes to the database, use a specific function for each iteration of the loop.

Parent topic: [Use CDC with Microsoft SQL Server databases \[page 914\]](#)

Related Information

[Limitations \[page 916\]](#)

[Data Services columns \[page 917\]](#)

[Change Tracking method \[page 923\]](#)

[Replication Server method \[page 926\]](#)

29.7.3.1 Creating a CDC datastore

Before you enable the Changed Data Capture (CDC) method, create a datastore connection to Microsoft SQL Server to import the metadata of a capture instance into the current repository.

Perform the following steps in SAP Data Services Designer.

1. Select **Project > New > Datastore**.

The datastore editor opens.

2. Enter a name for the datastore. Ensure that the name is unique and that it indicates the purpose of the datastore. Complete the remaining options as described in the following table:

Option	Value
<i>Datastore type</i>	<i>Database</i>
<i>Database type</i>	<i>Microsoft SQL Server</i>
<i>Database version</i>	<i>Microsoft</i> or newer version.
<i>Database Subtype</i>	Automatically populates with SQL Server on Premise.
CDC type	Select <i>Native CDC</i> from the CDC dropdown list.
CDC method	Select <i>CDC</i> from the dropdown list that appears under the CDC method dropdown list.
<i>Database server name</i>	Enter the applicable database server name.
<i>Database name</i>	Enter the applicable database name.
<i>User name</i>	Enter the name of the user who has permission to access the database.
<i>Password</i>	Enter the password for the specified user.

3. Click *OK* to save the datastore.

Use the new datastore connection to import the metadata of a capture instance into the current repository.

Task overview: [Changed Data Capture \(CDC\) method \[page 919\]](#)

Related Information

[Importing CDC metadata \[page 921\]](#)

[Configuring a source table using CDC \[page 921\]](#)

[Using CDC for data flows in a WHILE loop \[page 922\]](#)

[Defining a database datastore \[page 96\]](#)

29.7.3.2 Importing CDC metadata

Use the Changed Data Capture (CDC) datastore that you created for the capture instance to import metadata.

When CDC is enabled on a table, Microsoft SQL Server creates a capture instance. A table can have up to two capture instances with different names. Before you perform the following steps, create a CDC datastore following the steps in [Creating a CDC datastore \[page 920\]](#).

i Note

SAP Data Services displays the capture instance name (instead of the underlying table name). Therefore, when you import by name, the name must be the capture instance name (not the table name).

1. Right-click the CDC datastore in the [Datastore](#) tab of the object library and click [Import by Name](#) or [Search](#).

When you import CDC metadata, Data Services lists the capture instance name and not the table name.

- Select [Import by Name](#) when you know the capture instance name of the table
- Select [Search](#) when you know a part of the capture instance name.

2. Select the applicable table and click [Import](#).

The import process adds the table to the [Tables](#) node under the [Datastore](#) node in the object library. The import operation adds the following columns to the original table:

Column name	Data type	Source of column
DI_SEQUENCE_NUMBER	integer	Generated by Data Services
DI_OPERATION_TYPE	varchar(1)	Generated by Data Services
MSSQL_TRAN_SEQNO	varchar(256)	Supplied by Microsoft SQL Server
MSSQL_TRAN_TIMESTAMP	timestamp	Supplied by Microsoft SQL Server
MSSQL_COLUMN_UPDATE_MASK	varchar(258)	Supplied by Microsoft SQL Server

Task overview: [Changed Data Capture \(CDC\) method \[page 919\]](#)

Related Information

[Creating a CDC datastore \[page 920\]](#)

[Configuring a source table using CDC \[page 921\]](#)

[Using CDC for data flows in a WHILE loop \[page 922\]](#)

29.7.3.3 Configuring a source table using CDC

A source table is required when you use the Changed Data Capture (CDC) method for Microsoft SQL Server data.

Ensure that you perform the following tasks before you configure a source table using CDC:

- Create a datastore specifically for Changed Data Capture following instructions in [Creating a CDC datastore \[page 920\]](#).
 - Import the CDC change instance table following instructions in [Importing CDC metadata \[page 921\]](#).
1. Drag a Microsoft SQL Server capture instance table from the applicable SQL Server database datastore into a data flow in your workspace.
The table automatically becomes a source object.
 2. Double-click the source object in the data flow to open the source table editor.
 3. Click the [CDC Options](#) tab.
 4. Optional. Select [Enable check-point](#).
With a check point in place, the next time you run the CDC job it reads only the rows inserted into the CDC table since the last check-point.
 5. Optional. Select [Automatically delete rows after reading](#).
 6. Save your changes.

Task overview: [Changed Data Capture \(CDC\) method \[page 919\]](#)

Related Information

[Creating a CDC datastore \[page 920\]](#)

[Importing CDC metadata \[page 921\]](#)

[Using CDC for data flows in a WHILE loop \[page 922\]](#)

[CDC table source](#)

29.7.3.4 Using CDC for data flows in a WHILE loop

If a data flow runs in a WHILE loop to retrieve changes to the database, use a specific function for each iteration of the loop.

For the CDC method, check-points apply at the job level. Use the function `set_cdc_checkpoint()`, which instructs the source reader to set check-points so that the next loop iteration picks up the latest changes. Call this function for all datastores that you use in all the data flows of the job.

The following sample code shows the format of the function:

```
set_cdc_checkpoint('<datastore>');
```

The value `<datastore>` is the name of the datastore that contains the CDC tables.

❖ Example

```
set_cdc_checkpoint('MyCdcSource');
```

The function returns 1 if it successfully sets the check-point, otherwise the function returns 0.

For more information about the WHILE loop and the `set_cdc_checkpoint()` function, see the *Reference Guide*.

Parent topic: [Changed Data Capture \(CDC\) method \[page 919\]](#)

Related Information

[Creating a CDC datastore \[page 920\]](#)

[Importing CDC metadata \[page 921\]](#)

[Configuring a source table using CDC \[page 921\]](#)

29.7.4 Change Tracking method

The Change Tracking method identifies that rows in a table have changed but ignores how many times the row has changed or the values of any intermediate changes.

SAP Data Services retrieves only the latest data available. Therefore, change tracking is limited in the historical questions it can answer compared to the Changed Data Capture (CDC) method. However, there is far less storage overhead because Data Services is not capturing the changed data. In addition, a synchronous tracking mechanism used to track the changes has minimal overhead to operations.

To use the Change Tracking method, first enable it in the Microsoft SQL Server database and the tables within the database that you plan to track. Then create a Microsoft SQL Server datastore with Change Tracking enabled and import the applicable metadata.

Data Services records Change tracking information for modified rows. Data Services records the values of the primary key column from the tracked table with the change information to identify the rows that have changed. To obtain the latest data for those rows, Data Services uses the primary key column values to join the source table with the tracked table.

Information about the changes can include the type of operation that caused the change (INSERT, UPDATE, or DELETE) or the columns that were changed as part of an UPDATE operation.

The Change Tracking method is not available with versions older than Microsoft SQL Server 2008.

Parent topic: [Use CDC with Microsoft SQL Server databases \[page 914\]](#)

Related Information

[Limitations \[page 916\]](#)

[Data Services columns \[page 917\]](#)

[Changed Data Capture \(CDC\) method \[page 919\]](#)

[Replication Server method \[page 926\]](#)

29.7.4.1 Creating a Change Tracking datastore

To use the Change Tracking method for Microsoft SQL Server, create a datastore and enable the Change Tracking method.

Before you perform the following steps, ensure that you enable Change Tracking in the Microsoft SQL Server database and the tables within the database that you plan to track. For detail, see your Microsoft SQL Server documentation.

Perform the following steps in SAP Data Services Designer.

1. Select **Project** > **New** > **Datastore**.

The datastore editor opens.

2. Enter a name for the datastore. Ensure that the name is unique and that it indicates the purpose of the datastore. Complete the remaining options as described in the following table:

Option	Value
<i>Datastore type</i>	<i>Database</i>
<i>Database type</i>	<i>Microsoft</i>
<i>Database version</i>	<i>Microsoft SQL Server 2008</i> or newer version.
<i>Database Subtype</i>	Automatically populates with SQL Server on Premise.
CDC type	Select <i>Native CDC</i> from the CDC dropdown list.
CDC method	Select <i>Change tracking</i> from the dropdown list that appears under the CDC method dropdown list.
<i>Database server name</i>	Enter the applicable database server name.
<i>Database name</i>	Enter the applicable database name.
<i>User name</i>	Enter the name of the user who has permission to access the database.
<i>Password</i>	Enter the password for the specified user.

3. Click **OK** to save the datastore.

Use the new datastore connection to import table metadata into the current repository.

Related Information

[Defining a database datastore \[page 96\]](#)

29.7.4.2 Importing Change Tracking metadata

Use the Change Tracking database datastore to import table metadata.

Before you perform the steps to import metadata, enable Change Tracking in the Microsoft SQL Server database and the tables within the database that you plan to track. For details, see your Microsoft SQL Server documentation.

Then, create a Microsoft SQL Server database datastore that is configured for Change Tracking by following the steps in [Creating a Change Tracking datastore \[page 924\]](#).

Perform the following steps in SAP Data Services Designer.

1. Right-click the Change Tracking datastore in the *Datastore* tab of the object library and click *Import by Name* or *Search*.

When you import using the Change Tracking datastore, Data Services shows only the tables that you selected when you set up Microsoft SQL Server for Change Tracking in the External Data pane.

- Select *Import by Name* when you know the name of the table
- Select *Search* when you know a part of the table name.

2. Select the applicable table and click *Import*.

The import operation adds the following columns to the original table:

Column name	Data type	Source of column
DI_SEQUENCE_NUMBER	integer	Generated by Data Services
DI_OPERATION_TYPE	varchar(1)	Generated by Data Services
MSSQL_SYS_CHANGE_VERSION	decimal(19,0)	Supplied by Microsoft SQL Server
MSSQL_SYS_CHANGE_CREATION_VERSION	decimal(19,0)	Supplied by Microsoft SQL Server
MSSQL_SYS_CHANGE_CONTEXT_MASK	varchar(256)	Supplied by Microsoft SQL Server

29.7.4.3 Configuring a source table to use Change Tracking

A source table is required when you use the Change Tracking method for Microsoft SQL Server data.

Before you configure a source table to use Change tracking, ensure that you perform the following tasks:

- Configure your Microsoft SQL Server database and select tables for Change Tracking.
- Create a Change Tracking datastore.
- Import the Microsoft SQL Server table that you set up for Change Tracking.

Perform the following steps in SAP Data Services Designer.

1. Drag a Microsoft SQL Server Change Tracking table from the applicable SQL Server database datastore into a data flow in your workspace.

The table automatically becomes a source object.

2. Double-click the name of the source object to open the source table editor.
3. Click the *CDC Options* tab.

4. Specify a value for the *CDC subscription name*.
Data Services uses this name to track the last check-point internally in the repository.
5. Select *Enable check-point*.
Once a check-point is placed, the next time the Change Tracking job runs, it reads only the rows inserted into the Change Tracking table since the last check-point.

Related Information

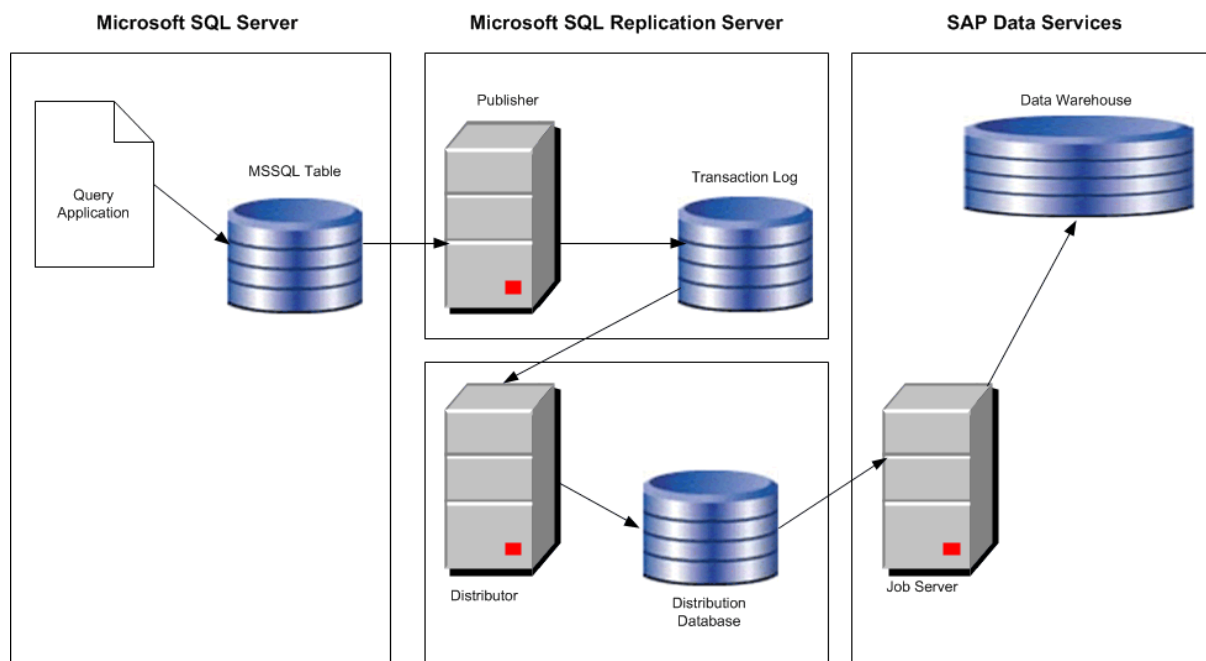
[CDC table source](#)

29.7.5 Replication Server method

The Replication Server method stores changes in a distribution database stored in the Replication Server.

How the process works

SAP Data Services obtains changed data from the Distribution database in the Microsoft SQL Replication Server. The following diagram shows how the changed data flows from the Replication Server to Data Services.



The process works as follows:

- An application changes a database and the Publisher within the Replication Server captures these changes within a transaction log.
- The Log Reader Agent in the Distributor reads the Publisher transaction log and saves the changed data in the Distribution database.
- The software reads the data from the command table within the Distribution database, applies appropriate filters, and creates input rows for a target data warehouse table.

Replication Server

The following terms describe the process and tools for a Microsoft SQL Server Replication Server.

Replication Server terminology

Term	Description
Article	A table, a partition, or a database object that the DBA specifies for replication. An article can be any of the following: • An entire table • Certain columns (using a vertical filter) • Certain rows (using a horizontal filter) • A stored procedure or view definition • The execution of a stored procedure • A view • An indexed view • A user-defined function
Distributor	A server that stores metadata, history data, and transactions into the distribution database. The software reads the distribution database to obtain changed data.
Publication	A collection of one or more articles from one database. A publication makes it easier to specify a logically related set of data and database objects that you want to replicate together.
Publisher	A server that makes data available for replication to other servers.
Subscriber	A server that receives replicated data. Subscribers subscribe to publications, not to individual articles within a publication. They subscribe only to the publications that they need, not to all of the publications available on a Publisher.

Distribution database tables

The following table contains descriptions for the tables that Data Services access within the Distribution database.

Tables in the Distribution database

Table	Description
MSarticles	Contains one row for each article that a Publisher replicates.
MSpublications	Contains one row for each publication that a Publisher replicates.
MSpublisher_databases	Contains one row for each Publisher and Publisher database pair that the local Distributor services.
MSrepl_commands	Contains rows of replicated commands (changes to data).

When you enable a database for replication, Replication Server creates tables on the source database. One of these tables is **Sysarticles** which contains a row for each article defined in this specific database. One of the columns in Sysarticles indicates what columns in a source table are being published.

Parent topic: [Use CDC with Microsoft SQL Server databases \[page 914\]](#)

Related Information

[Limitations \[page 916\]](#)

[Data Services columns \[page 917\]](#)

[Changed Data Capture \(CDC\) method \[page 919\]](#)

[Change Tracking method \[page 923\]](#)

29.7.5.1 Configuring the distribution database

SAP Data Services obtains changed data from the Distribution database in the Microsoft SQL Replication Server.

The changed data flows from the Replication Server to Data Services.

Before Data Services can connect to Microsoft SQL Server to extract data, you must configure the Distribution database in the Replication Server to capture changes to these tables.

The process is different between Microsoft SQL Server versions.

[Microsoft SQL Server 2000 \[page 929\]](#)

Configure the Distribution database in the Replication Server so SAP Data Services can connect to Microsoft SQL Server to extract data.

[Microsoft SQL Server 2005 and 2008 \[page 930\]](#)

Configure the Distribution database in the Replication Server so SAP Data Services can connect to Microsoft SQL Server to extract data.

29.7.5.1.1 Microsoft SQL Server 2000

Configure the Distribution database in the Replication Server so SAP Data Services can connect to Microsoft SQL Server to extract data.

For complete instructions to configure the Replication Server for Microsoft SQL Server 2000 databases, see your Microsoft SQL Enterprise Manager documentation.

The following steps summarize the procedure to configure the Replication Server for Microsoft SQL Server 2000 databases.

1. In the Microsoft SQL Enterprise Manager, open the Replication node and select the **Configure publishing, subscribers, and the Distribution** option.

A Wizard to create the Distributor and Distribution database opens. The following steps summarize the procedure to configure SQL Replication Server for your Microsoft SQL Server 2000 database.

- a. The Wizard generates distribution option components. Specify the values of the components in the **CDC** section of the *Advanced* datastore editor options when you create the CDC database datastore:
 - MSSQL distribution server name: Enter the value in *Distribution server name*.
 - MSSQL distribution database name: Enter the value in *Distribution database name*.
 - MSSQL distribution user name: Enter the value in *Distribution user name*.
 - MSSQL distribution password: Enter the value in *Distribution password*.

2. Select the **New Publications** option on the Replication node of the Microsoft SQL Enterprise Manager.

You create publications that specify the tables to publish under New Publications.

3. Make the following settings in the Advanced Options:
 - a. Select Transactional publication on the **Select Publication Type** window. Transactional publication updates data at the Publisher and send changes incrementally to the Subscriber.
 - b. In the **Commands** tab of the **Table Article Properties** window, select **XCALL** to create before images for the **UPDATE** and **DELETE** commands. Otherwise select **CALL**.
 - c. Clear the options **Create the stored procedures during initial synchronization of subscriptions** and **Send parameters in binary format** because the software does not use store procedures and has its own internal format.
 - d. On the **Snapshot** tab of the **Table Article Properties** window, select **Keep the existing table unchanged** because the software treats the table as a log.
 - e. Clear **Clustered indexes** because the software treats the table as a log and reads sequentially from it.
 - f. Specify a publication name and description. You specify this publication name in the *Publication name* option under the *CDC* group in the datastore editor when you define the CDC datastore.
 - g. Select the option **Yes, allow anonymous subscriptions** to save all transactions in the Distribution database.

Task overview: [Configuring the distribution database \[page 928\]](#)

Related Information

[Microsoft SQL Server 2005 and 2008 \[page 930\]](#)

29.7.5.1.2 Microsoft SQL Server 2005 and 2008

Configure the Distribution database in the Replication Server so SAP Data Services can connect to Microsoft SQL Server to extract data.

For complete instructions to configure the Replication Server for Microsoft SQL Server 2005 and 2008 databases, see your Microsoft SQL Enterprise Manager documentation.

The following steps summarize the procedure to configure the Replication Server for Microsoft SQL Server 2005 and 2008 databases.

1. Start the Microsoft SQL Server Management Studio.
2. Log in and go to the **Replication** node in the Object Explorer.
3. If this is the first time you configure distribution for this server, right-click the **Replication** node and select **Configure Distribution** in the context menu.
4. Right-click the **Replication** node again and select New > Publication. The **New Publication Wizard** opens.
5. Click **Next**.
6. Select the database to publish and click **Next**.
7. Under **Publication type**, select **Transactional publication** and click **Next**.
8. Click to select tables and columns to publish as articles. Then open **Article Properties**. For each selected table, click **Set Properties of Highlighted Table Article**. The **Article Properties** window opens:
 - a. Set the following options to False:
 - Copy clustered index
 - Copy INSERT, UPDATE and DELETE stored procedures
 - Create schemas at subscriber
 - b. Set the **Action if name is in use** to **Keep the existing table unchanged**.
 - c. Set **Update delivery format** and **Delete delivery format** to **XCALL< stored procedure>**.
 - d. Click **OK** to save the article properties.
9. Click **Next**.
10. Click **Add** to add row filters.
11. Click **Next** to skip filtering the data in your publication.
12. Configure Agent Security and specify the account connection setting:
 - a. For the Snapshot Agent, click **Security Settings**.
 - b. Specify the account under which the snapshot agent will run.
 - c. Configure the account with system administration privileges.
 - d. Specify the account that connects to the publisher.
 - e. Click **OK**.
 - f. For the Log Reader Agent, the software selects **Use the security settings from the Snapshot Agent** by default. To use different settings, clear this option and click **Security Settings**.

To set security settings, you have to log in as a user with system administration privileges.
13. In the **Wizard Actions** window, select **Create the publication** then click **Next**.
14. To complete the wizard, enter a Publication name and click **Finish**.

Task overview: [Configuring the distribution database \[page 928\]](#)

Related Information

[Microsoft SQL Server 2000 \[page 929\]](#)

29.7.5.2 Configure Data Services

Configure SAP Data Services to use the Replication Server CDC method for Microsoft SQL Server databases.

After you prepare the distribution database in Microsoft SQL Enterprise Server, prepare Data Services for the Replication Server CDC method. Configure Data Services to read and load changed data from SQL Server databases using the Replication Server. To configure Data Services for Replication Server CDC method, perform the following procedures:

- Create a CDC datastore
- Import metadata for Microsoft SQL Server tables
- Configure a CDC source

[Creating the Replication Server CDC datastore \[page 931\]](#)

Create a CDC database datastore so you can access the tables for change data capture processes.

[Importing CDC metadata \[page 932\]](#)

Use the Replication Server method CDC database datastore that you created to import the CDC metadata.

[Configuring a source table using replication \[page 933\]](#)

Configure a change data capture source table in a data flow for the Replication Server CDC method.

29.7.5.2.1 Creating the Replication Server CDC datastore

Create a CDC database datastore so you can access the tables for change data capture processes.

Perform the steps to configure the distribution database described in the section [Configuring the distribution database \[page 928\]](#).

Perform the following steps in SAP Data Services Designer:

1. Select **Project > New > Datastore**.

The datastore editor opens.

2. Enter a name for the datastore. Ensure that the name is unique and that it indicates the purpose of the datastore. Complete the remaining options as described in the following table:

Option	Value
<i>Datastore type</i>	<i>Database</i>
<i>Database type</i>	<i>Microsoft</i>
<i>Database version</i>	<i>Microsoft SQL Server 2008 or older version.</i>

Option	Value
<i>Database Subtype</i>	Automatically populates with <i>SQL Server on Premise</i> .
CDC type	Select <i>Native CDC</i> from the CDC dropdown list.
CDC method	Select <i>Replication</i> from the dropdown list that appears under the CDC method dropdown list.
<i>Database server name</i>	Enter the applicable database server name.
<i>Database name</i>	Enter the applicable database name.
<i>User name</i>	Enter the name of the user who has permission to access the database.
<i>Password</i>	Enter the password for the specified user.

3. In the *CDC* section under *Advanced*, enter the names that you created for this datastore when you configured the Distributor and Publisher in the Replication Server:

Find descriptions for the following options in [Microsoft SQL Server datastore \[page 129\]](#).

- Distribution server name
- Distribution database name
- Publication name
- Distribution user name
- Distribution password

4. Click *OK* to save the datastore.

Use the new CDC database datastore to import table metadata into the current repository.

Task overview: [Configure Data Services \[page 931\]](#)

Related Information

[Importing CDC metadata \[page 932\]](#)

[Configuring a source table using replication \[page 933\]](#)

[Defining a database datastore \[page 96\]](#)

29.7.5.2.2 Importing CDC metadata

Use the Replication Server method CDC database datastore that you created to import the CDC metadata.

Ensure that you perform the following tasks before you perform the steps to import CDC metadata:

- Configure the distribution database following the steps in [Configuring the distribution database \[page 928\]](#).
- Create the CDC database datastore following the steps in [Creating the Replication Server CDC datastore \[page 931\]](#).

After you create a CDC datastore, you can use it to import CDC table metadata. In the object library, right-click the datastore name and select [Open](#), [Import by Name](#), or [Search](#).

1. Right-click the CDC datastore in the [Datastore](#) tab of the object library and click [Import by Name](#) or [Search](#). You can also browse the metadata that appears in the External Metadata pane.

Only the CDC tables that you select when you set up Microsoft SQL Server for CDC are visible when you browse external metadata. Data Services uses the MSpublications and MSarticles table in the Distribution database of SQL Replication Server to create a list of published tables.

- Select [Import by Name](#) when you know the name of the table
- Select [Search](#) when you know a part of the table name.

2. Select the applicable table and click [Import](#).

When you import each CDC table, the software uses the Sysarticles table in the Publisher database of SQL Replication Server to display only published columns. The import operation adds the following columns to the original table:

Column name	Data type	Source of column
DI_SEQUENCE_NUMBER	integer	Generated by Data Services
DI_OPERATION_TYPE	varchar(1)	Generated by Data Services
MSSQL_TRAN_SEQNO	varchar(256)	Supplied by the Replication Server
MSSQL_TRAN_TIMESTAMP	timestamp	Supplied by the Replication Server

Task overview: [Configure Data Services \[page 931\]](#)

Related Information

[Creating the Replication Server CDC datastore \[page 931\]](#)

[Configuring a source table using replication \[page 933\]](#)

29.7.5.2.3 Configuring a source table using replication

Configure a change data capture source table in a data flow for the Replication Server CDC method.

Ensure that you perform the following tasks before you perform the steps to configure a CDC source table:

- Configure the distribution database following the steps in [Configuring the distribution database \[page 928\]](#).
- Create the CDC database datastore following the steps in [Creating the Replication Server CDC datastore \[page 931\]](#).
- Import the applicable replication table using the CDC datastore following the steps in [Importing CDC metadata \[page 932\]](#).

Perform the following steps in SAP Data Services Designer:

1. Drag a Microsoft SQL Server CDC table from the applicable SQL Server database datastore into a data flow in your workspace.

The table automatically becomes a source object.

2. Double-click the name of the source object to open the source table editor.
3. Click the *CDC Options* tab.
4. Specify a value for the *CDC subscription name*.

Data Services uses this name to track the last check-point internally in the repository.

5. Select *Enable check-point*.

For information about how check points work for the Replication Server method, see [Using check points with replication servers \[page 934\]](#).

Task overview: [Configure Data Services \[page 931\]](#)

Related Information

[Creating the Replication Server CDC datastore \[page 931\]](#)

[Importing CDC metadata \[page 932\]](#)

29.7.5.2.3.1 Using check points with replication servers

Select to use check points when you configure the source table.

A Log Reader Agent in the Microsoft SQL Replication Server reads the transaction log of the Publisher. The Log Reader Agent saves the changed data into the Distribution database, which SAP Data Services uses as the CDC data source. During processing, Data Services appends rows of changed data to the previous load in the CDC data source.

When you enable check points, a CDC job uses the CDC Subscription name to read the most recent set of appended rows and to mark the end of the read. If you do not enable check points, the CDC job reads all the rows in the CDC data source, which increases processing time.

See the instructions for configuring the source table for replication servers to set options for using check points.

29.7.5.2.3.2 Using before images from Microsoft SQL Server sources

The before image of an update row is the image before the software process changes the row.

Perform the following steps to capture before-image update rows when applicable:

1. Make sure the Replication Server is set up to retrieve full before images.
2. When you create a Publication in the Replication Server, specify **XCALL** for `UPDATE` commands and `DELETE` commands to obtain before images.
3. Select the *Get before-images for each update row* option in the CDC table source editor in SAP Data Services data flow.

After you check the *Get before-images for each update row* option, Data Services processes two rows for every update. In addition to the performance impact of this data volume increase, the before- and after-image pairs could be separated or lost depending on the design of your data flow. This mispairing could cause data integrity issues.

i Note

The Map_CDC_Operation transform can resolve integrity problems, but undesirable results can still occur because of programming errors. When you use functions and transforms that re-order, re-direct, eliminate, and multiply the number of rows in a data flow, be aware of the possible impact to targets.

Related Information

[Using before-images \[page 887\]](#)

29.8 Use CDC with timestamp-based sources

Use Timestamp-based CDC to track changes if you are using sources other than Oracle 9i, DB2 8.2, mainframes accessed through IBM II Classic Federation, or mainframes accessed through Attunity and if the following conditions are true:

- There are date and time fields in the tables being updated
- You are updating a large table that has a small percentage of changes between extracts and an index on the date and time fields
- You are not concerned about capturing intermediate results of each transaction between extracts (for example, if a customer changes regions twice in the same day).

You should not use the Timestamp-based CDC under the following circumstances:

- You have a large table, a large percentage of it changes between extracts, and there is no index on the timestamps.
- You need to capture physical row deletes.

- You need to capture multiple events occurring on the same row between extracts.

Generally, the term timestamp refers to date, time, or datetime values. In the case of timestamp-based CDC, the source table has either `CREATE` or `UPDATE` timestamps for each row.

Timestamps can indicate whether a row was created or updated. Some tables have both create and update timestamps; some tables have just one. In the case of timestamp-based CDC, tables contain at least one update timestamp.

Some systems have timestamps with dates and times, some with just the dates, and some with monotonically generated increasing numbers. In the case of timestamp-based CDC, dates and generated numbers are treated the same.

i Note

For the timestamps based on real time, time zones can become important. If you keep track of timestamps using the nomenclature of the source system (that is, using the source time or source-generated number), you can treat both temporal (specific time) and logical (time relative to another time or event) timestamps the same way.

Related Information

[Processing timestamps \[page 936\]](#)

[Overlaps \[page 938\]](#)

[Types of timestamps \[page 943\]](#)

29.8.1 Processing timestamps

The basic technique for using timestamps to determine changes and to save the highest timestamp loaded in a given job and start the next job with that timestamp.

To do this, create a status table that tracks the timestamps of rows loaded in a job. At the end of a job, `UPDATE` this table with the latest loaded timestamp. The next job then reads the timestamp from the status table and selects only the rows in the source for which the timestamp is later than the status table timestamp.

The following example illustrates the technique. Assume that the last load occurred at 2:00 PM on January 1, 1998. At that time, the source table had only one row (key=1) with a timestamp earlier than the previous load. SAP Data Services loads this row into the target table and updates the status table with the highest timestamp loaded: 1:10 PM on January 1, 1998. After 2:00 PM the software adds more rows to the source table:

Source table

Key	Data	Update_Timestamp
1	Alvarez	01/01/98 01:10 PM
2	Tanaka	01/01/98 02:12 PM

Key	Data	Update_Timestamp
3	Lani	01/01/98 02:39 PM

Target table

Key	Data	Update_Timestamp
1	Alvarez	01/01/98 01:10 PM

Status table

Last_Timestamp
01/01/98 01:10 PM

At 3:00 PM on January 1, 1998, the job runs again. This time the job does the following:

1. Reads the Last_Timestamp field from the status table (01/01/98 01:10 PM).
2. Selects rows from the source table whose timestamps are later than the value of Last_Timestamp. The SQL command to select these rows is:

```
SELECT * FROM Source
WHERE 'Update_Timestamp' > '01/01/98 01:10 pm'
```

This operation returns the second and third rows (key=2 and key=3).

3. Loads these new rows into the target table.
4. Updates the status table with the latest timestamp in the target table (01/01/98 02:39 PM) with the following SQL statement:

```
UPDATE STATUS SET 'Last_Timestamp' = SELECT MAX('Update_Timestamp') FROM
target_table
```

The target shows the new data:

Source table

Key	Data	Update_Timestamp
1	Alvarez	01/01/98 01:10 PM
2	Tanaka	01/01/98 02:12 PM
3	Lani	01/01/98 02:39 PM

Target table

Key	Data	Update_Timestamp
2	Tanaka	01/01/98 02:12 PM
3	Lani	01/01/98 02:39 PM

Key	Data	Update_Timestamp
1	Alvarez	01/01/98 01:10 PM

Status table

Last_Timestamp
01/01/98 02:39 PM

To specify these operations, a data flow requires the following objects (and assumes all the required metadata for the source and target tables has been imported):

- A data flow to extract the changed data from the source table and load it into the target table, such as Source > Query > Target.
The query selects rows from SOURCE_TABLE to load to TARGET_TABLE.
For example:

```
SOURCE.UPDATE_TIMESTAMP > $Last_Update
```

The query includes a where clause to filter rows between timestamps.

- A work flow to perform the following:
 1. Read the status table
 2. Set the value of a variable to the last timestamp
 3. Call the data flow with the variable passed to it as a parameter
 4. Update the status table with the new timestamp
- A job to execute the work flow

29.8.2 Overlaps

Unless source data is rigorously isolated during the extraction process (which typically is not practical), there is a window of time when changes can be lost between two extraction runs. This overlap period affects source-based changed-data capture because this kind of data capture relies on a static timestamp to determine changed data.

For example, suppose a table has 1000 rows (ordered 1 to 1000). The job starts with timestamp 3:00 and extracts each row. While the job is executing, it updates two rows (1 and 1000) with timestamps 3:01 and 3:02, respectively. The job extracts row 200 when someone updates row 1. When the job extracts row 300, it updates row 1000. When complete, the job extracts the latest timestamp (3:02) from row 1000 but misses the update to row 1.

Here is the data in the table:

Row Number	Column A
1	...
2	...

Row Number	Column A
3	...
...	...
200	...
...	...
600	...
...	...
1000	...

There are three techniques for handling this situation:

- Overlap avoidance
- Overlap reconciliation
- Presampling

The following sections describe these techniques and their implementations in SAP Data Services. This section continues on the assumption that there is at least an update timestamp.

Related Information

[Overlap avoidance \[page 939\]](#)

[Overlap reconciliation \[page 939\]](#)

[Presampling \[page 940\]](#)

[Types of timestamps \[page 943\]](#)

29.8.2.1 Overlap avoidance

In some cases, it is possible to set up a system where there is no possibility of an overlap. You can avoid overlaps if there is a processing interval where no updates are occurring on the target system.

For example, if you can guarantee that the data extraction from the source system does not last more than one hour, you can run a job at 1:00 AM every night that selects only the data updated the previous day until midnight. While this regular job does not give you up-to-the-minute updates, it guarantees that you never have an overlap and greatly simplifies timestamp management.

29.8.2.2 Overlap reconciliation

Overlap reconciliation requires a special extraction process that reapplies changes that could have occurred during the overlap period. This extraction can be executed separately from the regular extraction. For example, if the highest timestamp loaded from the previous job was 01/01/98 10:30 PM and the overlap period is one hour, overlap reconciliation reapplies the data updated between 9:30 PM and 10:30 PM on January 1, 1998.

The overlap period is usually equal to the maximum possible extraction time. If it can take up to **<n>** hours to extract the data from the source system, an overlap period of **<n>** (or **<n>** plus some small increment) hours is recommended. For example, if it takes at most two hours to run the job, an overlap period of at least two hours is recommended.

There is an advantage to creating a separate overlap data flow. A "regular" data flow can assume that all the changes are new and make assumptions to simplify logic and improve performance. For example, rows flagged as INSERT are often loaded into a fact table, but rows flagged as UPDATE rarely are. Thus, the regular data flow selects the new rows from the source, generates new keys for them, and uses the database loader to add the new facts to the target database. Because the overlap data flow is likely to apply the same rows again, it cannot blindly bulk load them or it creates duplicates. Therefore, the overlap data flow must check whether the rows exist in the target and insert only the ones that are missing. This lookup affects performance; therefore, perform it for as few rows as possible.

If the data volume is sufficiently low, you can load the entire new data set using this technique of checking before loading, avoiding the need to create two different data flows.

29.8.2.3 Presampling

Presampling eliminates the overlap by first identifying the most recent timestamp in the system, saving it, and then extracting rows up to that timestamp.

The technique is an extension of the simple timestamp processing technique. The main difference is that the status table now contains a start and an end timestamp. The start timestamp is the latest timestamp extracted by the previous job; the end timestamp is the timestamp selected by the current job.

To return to the example: The last extraction job loaded data from the source table to the target table and updated the status table with the latest timestamp loaded:

Source table

Key	Data	Update_Timestamp
1	Alvarez	01/01/98 01:10 PM
2	Tanaka	01/01/98 02:12 PM
3	Lani	01/01/98 02:39 PM

Target table

Key	Data	Update_Timestamp
1	Alvarez	01/01/98 01:10 PM

Status table

Start_Timestamp	End_Timestamp
01/01/98 01:10 PM	NULL

Now it's 3:00 PM on January 1, 1998, and the next job runs; it does the following:

1. Selects the most recent timestamp from the source table and inserts it into the status table as the End Timestamp.

The SQL command to select one row is:

```
SELECT MAX(Update_Timestamp) FROM source table
```

The status table becomes:

Status table

Start_Timestamp	End_Timestamp
01/01/98 01:10 PM	01/01/98 02:39 PM

1. Selects rows from the source table whose timestamps are greater than the start timestamp but less than or equal to the end timestamp. The SQL command to select these rows is:

```
SELECT *
FROM source table
WHERE Update_Timestamp > '1/1/98 1:10pm'
AND Update_Timestamp <= '1/1/98 2:39pm'
```

This operation returns the second and third rows (key=2 and key=3)

2. Loads these new rows into the target table.
3. Updates the status table by setting the start timestamp to the previous end timestamp and setting the end timestamp to NULL.

The table values end up as follows:

Source table

Key	Data	Update_Timestamp
1	Alvarez	01/01/98 01:10 PM
2	Tanaka	01/01/98 02:12 PM
3	Lani	01/01/98 02:39 PM

Target table

Key	Data	Update_Timestamp
1	Alvarez	01/01/98 01:10 PM
2	Tanaka	01/01/98 02:12 PM
3	Lani	01/01/98 02:39 PM

Status table

Start_Timestamp	End_Timestamp
01/01/98 02:39 PM	NULL

To enhance the previous example to consider the overlap time requires the following changes to the work flow:

- A data flow to extract the changes since the last update and before the most recent timestamp, such as Source > Query > Target.

The query selects rows from SOURCE_TABLE to load to TARGET_TABLE.

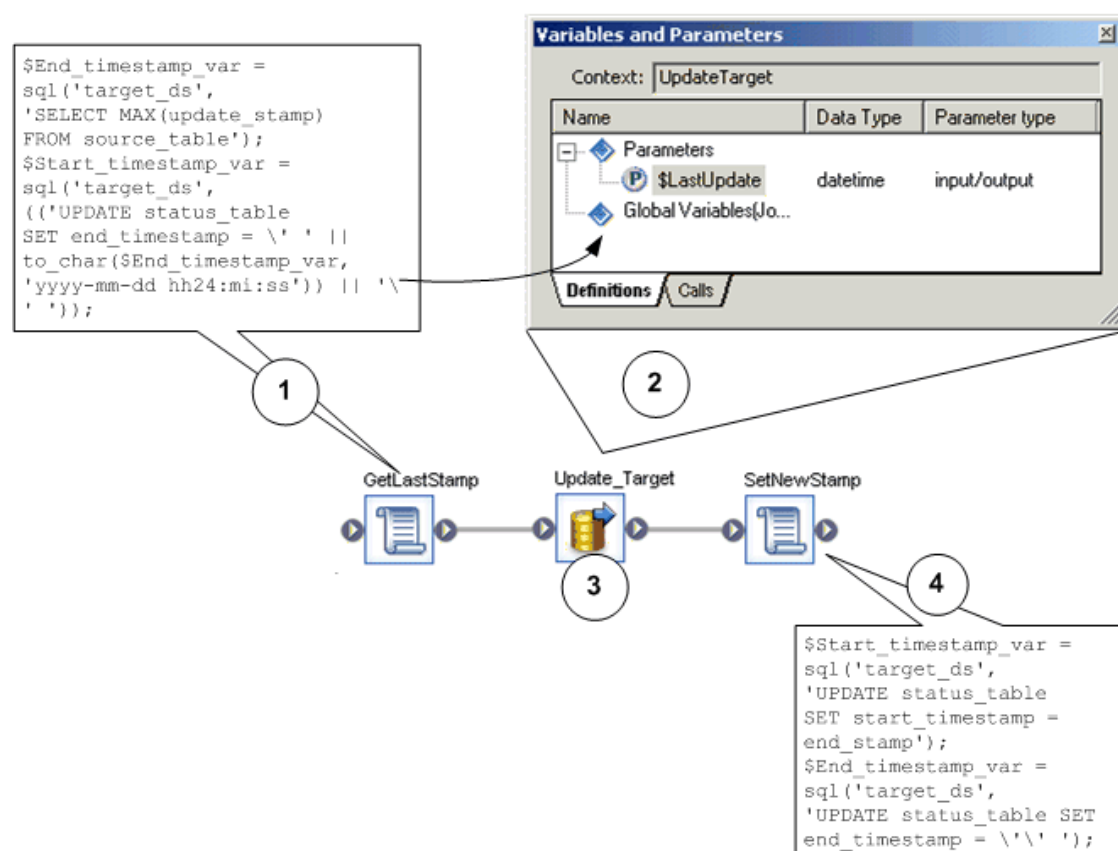
For example:

```
SOURCE.UPDATE_TIMESTAMP > $start_last_update and
SOURCE.UPDATE_TIMESTAMP < $end_last_update
```

The query includes a where clause to filter rows between timestamps.

- A work flow to perform the following:
 1. Read the source table to find the most recent timestamp.
 2. Set the value of two variables to the start of the overlap time and to the end of the overlap time, respectively.
 3. Call the data flow with the variables passed to it as parameters.
 4. Update the start timestamp with the value from end timestamp and set the end timestamp to NULL.

Work flow: Changed data with timestamps



Related Information

[Processing timestamps \[page 936\]](#)

29.8.3 Types of timestamps

Some systems have timestamps that record only when rows are created. Others have timestamps that record only when rows are updated. (Typically, update-only systems set the update timestamp when the row is created or updated.) Finally, there are systems that keep separate timestamps that record when rows are created and when they are updated.

Related Information

[Create-only timestamps \[page 943\]](#)

[Update-only timestamps \[page 943\]](#)

[Create and update timestamps \[page 944\]](#)

29.8.3.1 Create-only timestamps

If the source system provides only create timestamps, you have these options:

- If the table is small enough, you can process the entire table to identify the changes. The section [Use CDC for targets \[page 950\]](#), describes how to identify changes.
- If the table never gets updated, you can extract only the new rows.
- If the table is large and gets updated, you can combine the following two techniques:
 - Periodically (for example, daily) extract only the new rows.
 - Less frequently (for example, weekly) extract the updated rows by processing the entire table.

29.8.3.2 Update-only timestamps

Using only an update timestamp helps minimize the impact on the source systems, but it makes loading the target systems more difficult. If the system provides only an update timestamp and there is no way to tell new rows from updated rows, your job has to reconcile the new data set against the existing data using the techniques described in the section [Use CDC for targets \[page 950\]](#).

29.8.3.3 Create and update timestamps

Both timestamps allow you to easily separate new data from updates to the existing data. The job extracts all the changed rows and then filters unneeded rows using their timestamps.

Accomplish these extractions in the software by adding the WHERE clause from the following SQL commands into an appropriate query transform:

- Find new rows:

```
SELECT * FROM source_table
WHERE Create_Timestamp > $Last_Timestamp
```

- Find updated rows:

```
SELECT * FROM source_table
WHERE Create_Timestamp <= $Last_Timestamp AND
Update_Timestamp > $Last_Timestamp)
```

From here, the new rows go through the key-generation process and are inserted into the target, and the updated rows go through the key-lookup process and are updated in the target.

For performance reasons, you might want to separate the extraction of new rows into a separate data flow to take advantage of bulk loading into the target. The updated rows cannot be loaded by bulk into the same target at the same time.

29.8.4 Timestamp-based CDC examples

29.8.4.1 Preserving generated keys

For performance reasons, many data warehouse dimension tables use generated keys to join with the fact table. For example, customer ABC has a generated key 123 in the customer dimension table. All facts for customer ABC have 123 as the customer key. Even if the customer dimension is small, you cannot simply reload it every time a record changes: unless you assign the generated key of 123 to the customer ABC, the customer dimension table and the fact tables do not correlate.

You can preserve generated keys by either using the lookup function or comparing tables.

Related Information

[Using the lookup function \[page 945\]](#)

[Comparing tables \[page 946\]](#)

29.8.4.1.1 Using the lookup function

If history preservation is not an issue and the only goal is to generate the correct keys for the existing rows, the simplest technique is to look up the key for all rows using the `lookup` function in a query. If you do not find the key, generate a new one.

In the following example, the customer dimension table contains generated keys. When you run a job to update this table, the source customer rows must match the existing keys.

Source customer table

Company Name	Customer ID
ABC	001
DEF	002
GHI	003
JKL	004

Target dimension table

Gen_Key	Company Name	Customer ID
123	ABC	001
124	DEF	002
125	GHI	003

This example data flow does the following:

1. Extracts the source rows.
2. Retrieves the existing keys using a `lookup` function in the mapping of a new column in a query.
3. Loads the result into a file (to be able to test this stage of the data flow before adding the next steps).

The `lookup` function compares the source rows with the target. The arguments for the function are as follows:

lookup function arguments	Description
<code>target_ds.owner.customer</code>	Fully qualified name of the target table containing the generated keys.
<code>GKey</code>	The column name in the target table containing the generated keys.
<code>NULL</code>	<code>NULL</code> value to insert in the key column if no existing key is found.

lookup function arguments	Description
'PRE_LOAD_CACHE'	Caching option to optimize the lookup performance.
Customer_ID	The column in the target table containing the value to use to match rows.
Customer_ID	The column in the source table containing the values to use to match rows.

The resulting data set contains all the rows from the source with generated keys where available:

Result data set

Gen_Key	Company Name	Customer ID
123	ABC	001
124	DEF	002
125	GHI	003
NULL	JKL	004

Adding a new generated key to the new records requires filtering out the new rows from the existing and updated rows. In the data flow, this requires the following steps: A query to select the rows with NULL generated keys. A Key_Generation transform to determine the appropriate key to add. A target to load the new rows into the customer dimension table.

This data flow handles the new rows; however, the rows from the source whose keys were found in the target table might contain updated data. Because this example assumes that preserving history is not a requirement, the software loads all rows from the source into the target.

29.8.4.1.1.1 Handling updated rows in the data flow

The data flow requires new steps to handle updated rows, as follows:

1. A new line leaving the query that looked up the existing keys.
2. A query to filter the rows with existing keys from the rows with no keys.
3. A target to load the rows into the customer dimension table.

29.8.4.1.2 Comparing tables

The drawback of the generated-keys method is that even if the row has not been changed, it generates an UPDATE and is loaded into the target. If the amount of data is large, a table-comparison transform provides a better alternative by allowing the data flow to load only changed rows.

The table-comparison transform examines all source rows and performs the following operations:

- Generates an INSERT for any new row not in the target table.
- Generates an UPDATE for any row in the target table that has changed.
- Ignores any row that is in the target table and has not changed.
- Fills in the generated key for the updated rows.

You can then run the result through the key-generation transform to assign a new key for every INSERT. This is the data set that the software loads into the target table.

The data flow that accomplishes this transformation includes the following steps:

1. A source to extract the rows from the source table(s).
2. A query to map columns from the source.
3. A table-comparison transform to generate INSERT and UPDATE rows and to fill in existing keys.
4. A key-generation transform to generate new keys.
5. A target to load the rows into the customer dimension table.

29.8.4.2 Preserving history

History preserving allows the data warehouse or data mart to maintain the history of data so you can analyze it over time. Most likely, you will perform history preservation on dimension tables.

For example, if a customer moves from one sales region to another, simply updating the customer record to reflect the new region would give you misleading results in an analysis by region over time because all purchases made by a customer before the move would incorrectly be attributed to the new region.

SAP Data Services provides a special transform that preserves data history to prevent this kind of situation. The History_Preserving transform ignores everything but rows flagged as UPDATE. For these rows, it compares the values of specified columns and, if the values have changed, flags the row as INSERT. This produces a second row in the target instead of overwriting the first row.

To expand on how the software would handle the example of the customer who moves between regions:

- If `Region` is a column marked for comparison, the History_Preserving transform generates a new row for that customer.
- A Key_Generation transform gives the new row a new generated key and loads the row into the customer dimension table.
- The original row describing the customer remains in the customer dimension table with a unique generated key.

In the following example, one customer moved from the East region to the West region, and another customer's phone number changed.

Source Customer table

Customer	Region	Phone
Fred's Coffee	East	(212) 123-4567
Jane's Donuts	West	(650) 222-1212

Customer	Region	Phone
Sandy's Candy	Central	(115) 231-1233

Target Customer table

GKey	Customer	Region	Phone
1	Fred's Coffee	East	(212) 123-4567
2	Jane's Donuts	East	(201) 777-1717
3	Sandy's Candy	Central	(115) 454-8000

In this example, the data flow preserves the history for the `Region` column but does not preserve history for the `Phone` column. The data flow contains the following steps:

1. A source to extract the rows from the source table(s).
2. A query to map columns from the source.
3. A table-comparison transform to generate INSERTs and UPDATEs and to fill in existing keys.
4. A History_Preserving transform to convert certain UPDATE rows to INSERT rows.
5. A key-generation transform to generate new keys for the updated rows that are now flagged as INSERT.
6. A target to load the rows into the customer dimension table.

Now that there are two rows for Jane's Donuts, correlations between the dimension table and the fact table must use the highest key value.

Note that updates to non-history preserving columns update all versions of the row if the update is performed on the natural key (for example, `Customer`), and only update the latest version if the update is on the generated key (for example, `GKey`). You can control which key to use for updating by appropriately configuring the loading options in the target editor.

29.8.4.2.1 valid_from date and valid_to date

To support temporal queries like "What was the customer's billing address on May 24, 1998," SAP Data Services supports *Valid from* and *Valid to* date columns.

In history-preserving techniques, there are multiple records in the target table with the same source primary key values. A record from the source table is considered valid in the dimension table for all date values t such that the *Valid from* date is less than or equal to t , which is less than the *Valid to* date. (Valid in this sense means that the record's generated key value is used to load the fact table during this time interval.)

When you specify the *Valid from* and *Valid to* entries, the History_Preserving transform generates an UPDATE record before it generates an INSERT statement for history-preservation reasons (it converts an UPDATE into an INSERT). The UPDATE record will set the *Valid to* date column on the current record (the one with the same primary key as the INSERT) to the value in the *Valid from* date column in the INSERT record.

29.8.4.2.2 Update flag

To support slowly changing dimension techniques, Data Services enables you to set an update flag to mark the current record in a dimension table.

Value *Set value* in column *Column* identifies the current valid record in the target table for a given source table primary key.

When you specify *Column*, the History_Preserving transform generates an UPDATE record before it generates an INSERT statement.

This UPDATE record will set the *Column* value to *Reset value* in the target table record with the same source primary key as the INSERT statement.

In the INSERT statement the *Column* will be set to *Set value*.

When you specify entries in both the groups, the History_Preserving transform generates only one extra UPDATE statement for every INSERT statement it produces. This UPDATE statement updates the *Valid to* value.

29.8.5 Additional job design tips

When designing a job to implement changed-data capture (CDC), you must consider:

- Synchronizing header and detail
- Capturing physical deletions

29.8.5.1 Synchronizing header and detail

Typically, source systems keep track of header and detail information changes in an independent way. For example, if a line-item status changes, its "last modified date" column updates, but the same column at the order header level does not update. Conversely, a change to the default ship-to address in the order header might impact none of the existing line items.

In some instances, however, your source system might not consistently update those tracking columns, or you might not have access to such information (for example, when rows are physically deleted). In these cases, you might choose to extract all header and detail information whenever any changes occur at the header level or in any individual line item.

To extract all header and detail rows when any of these elements have changed, use logic similar to this SQL statement:

```
SELECT ... FROM HEADER, DETAIL WHERE HEADER.ID = DETAIL.ID AND (HEADER.LAST_MODIFIED  
BETWEEN $G_SDATE AND $G_EDATE OR DETAIL.LAST_MODIFIED BETWEEN $G_SDATE AND $G_EDATE)
```

For some databases, this WHERE clause is not well optimized and might cause serious performance degradation. You might opt to relax that clause by removing one of the upper bounds, such as in:

```
... WHERE HEADER.ID = DETAIL.ID AND (HEADER.LAST_MODIFIED BETWEEN $G_SDATE AND $G_EDATE OR  
DETAIL.LAST_MODIFIED >= $G_SDATE) ...
```

This might retrieve a few more rows than originally intended, but it might improve the final performance of your system while not altering the result of your target database.

29.8.5.2 Capturing physical deletions

When your source system allows rows to be physically deleted, your job should include logic to update your target database correspondingly. There are several ways to do this:

- Scan a log of operations — If your system logs transactions in a readable format or if you can alter the system to generate such a log, then you can scan that log to identify the rows you need to delete.
- Perform a full refresh — Simply reload all of the data, therefore fully synchronizing the source system and the target database.
- Perform a partial refresh based on a data-driven time-window — For example, suppose that the source system only allows physical deletion of orders that have not been closed. If the first non-closed order in your source table occurred six months ago, then by refreshing the last six months of data you are guaranteed to have achieved synchronization.
- Perform a partial refresh based on a business-driven time-window — For example, suppose that the business that the job supports usually deletes orders shortly after creating them. In this case, refreshing the last month of orders is appropriate to maintain integrity.
- Check every order that could possibly be deleted — You must verify whether any non-closed order has been deleted. To be efficient, this technique requires you to keep a record of the primary keys for every object that is a candidate for deletion.

When physical deletions of detail information in a header-detail relationship are possible (for example, removing line items from an existing order), then you must capture these physical deletions when synchronizing header and detail information.

29.9 Use CDC for targets

Source-based changed-data capture is almost always preferable to target-based capture for performance reasons.

Some source systems, however, do not provide enough information to make use of the source-based changed-data capture techniques. Target-based changed-data capture allows you to use the technique when source-based change information is limited.

30 Data Services data types

SAP Data Services supports several specific data types.

The following table provides a list of the data types that Data Services supports with a brief description. For complete information about the data types, see the *Reference Guide*.

Supported data types

Data type	Description
blob	Stores binary data. Use for multimedia data such as images, audio, and video. Do not use columns that contain blob data types in comparisons, calculations, or data type conversions. Blob data type has many of the same limitations as the long data type.
date	Defines calendar dates. Specify date formats by generating strings using certain date format codes. For example, DD = 2-digit day of the month. MM = 2-digit month number.
datetime	Defines calendar dates and times. Behaves like a concatenation of date and time data items. Specify date rules for the date part and time rules for the time part. For example: dd:mm:yy hh24:mi:ss.
decimal	Defines exact decimal numbers. Specify the precision and scale when you specify a decimal data type: • Precision: Total number of digits in the value Use the following relation: $1 \leq \text{precision} \leq 96$ • Scale: Number of digits to the right of the decimal point. Use the following relation: $0 \leq \text{scale} \leq \text{precision}$
double	Defines an 8-byte floating point value. The 8-byte floating point value has radix, exponent range, and precision of the platform on which Data Services is running.

Data type	Description
int	Defines a 4-byte signed binary integer. Can have a plus or minus sign that indicates a positive or negative value. Sign can appear before or after the value with any number of blanks between the value and the sign. When the number does not have a plus or minus sign, the software considers it as positive.
interval	Defines differences between dates or times. Value is in days unless you specify another unit. Use the following conversion functions to make interval values available: <code>interval_to_char</code> <code>num_to_interval</code>
long	Stores character data of variable length up to 2 gigabytes. Use to represent character-based large objects or "clob". Converts several other database-specific large object types such as longvarchar and text, to the long data type. Supports mapping from all databases. Long data type has many of the same limitations as the blob data type.
numeric	Identical to the decimal data type.
real	Defines a 4-byte floating point value with radix, exponent range, and precision of the platform on which Data Services is running. Can have a plus or minus sign that indicates a positive or negative value. Sign can appear before or after the value with any number of blanks between the value and the sign. When the number does not have a plus or minus sign, the software considers it as positive. Databases store real values as a 32-bit approximation of the number. The approximation makes comparison results unpredictable when you use a real value in an equality or inequality comparison. Therefore, we recommend that you do not use a real value in a <code>WHERE</code> clause.

Data type	Description
time	Defines times of the day with no calendar date. Manages time operations in the format used by the database manager. Conversion operations to or from time require you to specify the format of the time value. Specify a time format by generating a string using codes and literal strings or punctuation. For example HH24 = 2-digit hour of the day.
timestamp	Supports the Oracle 9i timestamp data type with no zone information. Incorporates up to a 9-digit subsecond. Add or subtract timestamp values. The resulting data type from addition or subtraction operations depends on the operation and data types involved.
varchar	Can contain letters and numbers. Indicate the length when specifying a varchar data type.

Related Information

[Data Types](#)

31 Data type conversion

SAP Data Services converts data types during certain processes so that the data types from different sources use the same data types for processing.

Data Services converts data types under the following situations:

- Performs date arithmetic
- Converts to or from internal data types
- Processes expressions
- Works with number data types
- Converts between explicit data types
- Converts between native data types

[Date arithmetic \[page 954\]](#)

SAP Data Services performs implicit data type conversions when it performs arithmetic on date-type data.

[Unsupported data types \[page 955\]](#)

SAP Data Services reads, load, and calls stored procedures with unknown data types under certain circumstances.

[Conversion to or from internal data types \[page 956\]](#)

SAP Data Services uses internal conversion functions for converting incoming and outgoing metadata.

[Conversion of data types within expressions \[page 984\]](#)

When possible, SAP Data Services optimizes data flows by pushing expressions down to an underlying database manager.

[Conversion of number data types in expressions \[page 985\]](#)

SAP Data Services uses a type-promotion algorithm to evaluate expressions that contain more than one number data type.

[Conversion between explicit data types \[page 988\]](#)

Use function to convert data types between explicit data types.

[Conversion between native data types \[page 988\]](#)

A Data Quality transform can get and set field data in a format other than the declared data type.

31.1 Date arithmetic

SAP Data Services performs implicit data type conversions when it performs arithmetic on date-type data.

Data Services uses implicit data type conversions when it performs date arithmetics on the following date types:

- date
- time

- datetime
- timestamp
- interval

The following table contains the data type returned after Data Services performs the stated operation.

Returned data types per operation

Operation	Return type
DATE + INTERVAL	DATE
TIME + INTERVAL	TIME
DATETIME + INTERVAL	DATETIME
INTERVAL + INTERVAL	INTERVAL
DATE - DATE	INTERVAL
DATE - INTERVAL	DATE
TIME - TIME	INTERVAL
TIME - INTERVAL	TIME
INTERVAL - INTERVAL	INTERVAL
DATETIME - DATE	INTERVAL
DATETIME - TIME	INTERVAL
DATETIME - DATETIME	INTERVAL
DATETIME - INTERVAL	DATETIME
TIMESTAMP + TIMESTAMP	INTERVAL
TIMESTAMP - TIMESTAMP	INTERVAL
TIMESTAMP + INTERVAL	TIMESTAMP
TIMESTAMP - INTERVAL	TIMESTAMP

31.2 Unsupported data types

SAP Data Services reads, load, and calls stored procedures with unknown data types under certain circumstances.

Data Services can work with unknown data types provided your database servers can do the following:

- Convert from VARCHAR to the native (unknown) data type
- Convert from the native (unknown) data type to VARCHAR

i Note

If the native database does not support a specific process, Data Services might have a problem loading VARCHAR to a physical CLOB column. For example, bulk loading or auto-correct load could fail if the database doesn't support those features.

When the software encounters a column assigned to an unsupported data type, it does not import the metadata for the column and issues an error. The file `errorlog.txt` contains the error information and the column name. To include the column in your job, convert the data type in the source to a supported data type. Reimport the metadata for the table and reprocess the job.

i Note

Use the `varchar_to_long` function to convert a VARCHAR data type to a LONG datatype before loading physical CLOB. If you are using a VARCHAR column in the physical schema for loading in a prior installation, this conversion still works.

31.3 Conversion to or from internal data types

SAP Data Services uses internal conversion functions for converting incoming and outgoing metadata.

Data Services converts data types under the following circumstances:

- Imports metadata from external sources or targets into the repository
- Loads data into an external table or file

After conversion, Data Services uses the internal data types instead of the datatypes that are specific to the database or application of the source data.

For template tables or Data_Transfer tables as targets, Data Services converts data from internal types to data types of the respective database management systems.

For information about data conversions for the following database types, consult the listed resource.

Databases documented elsewhere

Database	Resource
Redshift	<i>Data Services Supplement for Big Data</i>
Apache Hive	<i>Data Services Supplement for Big Data</i> <i>Supplement for Hadoop</i>
HP Vertica	<i>Data Services Supplement for Big Data</i>
SAP HANA	<i>Data Services Supplement for Big Data</i>
SAP Vora	<i>Data Services Supplement for Big Data</i>

[Attunity Streams \[page 958\]](#)

When you import metadata from an Attunity Streams source or target into the repository, SAP Data Services converts data types.

[Cobol copybook \[page 959\]](#)

When you import metadata from a COBOL copybook source or target into the repository, SAP Data Services converts data types.

[Data Federator \[page 960\]](#)

When you import metadata from a Data Federator source into the repository, SAP Data Services converts data types.

[IBM DB2 \[page 961\]](#)

When you import metadata from an IBM DB2 source or target into the repository, SAP Data Services converts data types.

[Informix \[page 963\]](#)

When you import metadata from a Informix source or target into the repository, SAP Data Services converts data types.

[Microsoft Excel \[page 965\]](#)

When you import metadata from a Microsoft Excel source or target into the repository, SAP Data Services converts data types.

[Microsoft SQL Server \[page 966\]](#)

When you import metadata from a Microsoft SQL Server source or target into the repository, SAP Data Services converts data types.

[MySQL \[page 969\]](#)

When you import metadata from a MySQL source or target into the repository, SAP Data Services converts data types.

[Netezza \[page 971\]](#)

When you import metadata from a Netezza source or target into the repository, SAP Data Services converts data types.

[ODBC \[page 972\]](#)

When you import metadata from an ODBC source or target into the repository, SAP Data Services converts data types.

[Oracle \[page 974\]](#)

When you import metadata from an Oracle source or target into the repository, SAP Data Services converts data types.

[SQL Anywhere \[page 976\]](#)

When you import metadata from a SQL Anywhere source or target into the repository, SAP Data Services converts data types.

[SAP ASE \[page 979\]](#)

When you import metadata from an SAP ASE source or target into the repository, SAP Data Services converts data types.

[SAP Sybase IQ \[page 980\]](#)

When you import metadata from an SAP Sybase IQ source or target into the repository, SAP Data Services converts data types.

[Teradata \[page 982\]](#)

When you import metadata from a Teradata source or target into the repository, SAP Data Services converts data types.

Related Information

[Amazon Redshift data types](#)

[Hive data type conversion](#)

[HP Vertica data type conversion](#)
[SAP HANA datatypes](#)
[SAP Vora data type conversions](#)

31.3.1 Attunity Streams

When you import metadata from an Attunity Streams source or target into the repository, SAP Data Services converts data types.

The following table shows the conversion from Attunity Streams data types to Data Services data types.

Data Services conversion of Attunity Streams data types

Attunity Streams data type	Converts to Data Services data type
bigint	double
i Note Because int is only four bytes, data is lost during the conversion.	
double	double
float	double
long varchar	Converted to long only if imported using an ODBC datastore.
real	real
date	datetime
decimal	decimal
integer	int
smallint	int
time	time
timestamp	datetime
varchar	varchar

Parent topic: [Conversion to or from internal data types \[page 956\]](#)

Related Information

[Cobol copybook \[page 959\]](#)
[Data Federator \[page 960\]](#)
[IBM DB2 \[page 961\]](#)
[Informix \[page 963\]](#)
[Microsoft Excel \[page 965\]](#)
[Microsoft SQL Server \[page 966\]](#)

[MySQL \[page 969\]](#)
[Netezza \[page 971\]](#)
[ODBC \[page 972\]](#)
[Oracle \[page 974\]](#)
[SQL Anywhere \[page 976\]](#)
[SAP ASE \[page 979\]](#)
[SAP Sybase IQ \[page 980\]](#)
[Teradata \[page 982\]](#)

31.3.2 Cobol copybook

When you import metadata from a COBOL copybook source or target into the repository, SAP Data Services converts data types.

The following table shows the conversion from COBOL copybook data types to Data Services data types.

Data Services conversion of COBOL copybooks data types

COBOL copybook data type	Converts to Data Services data type
computational	decimal
comp	decimal
computational-1	float
comp-1	float
computational-2	double
comp-2	double
computational-3	decimal
comp-3	decimal
computational-4	decimal
comp-4	decimal
computational-5	decimal
comp-5	decimal
computational-6	decimal
comp-6	decimal
computational-x	decimal
comp-x	decimal
binary	decimal
packed-decimal	decimal
float	float
double	double

COBOL copybook data type	Converts to Data Services data type
signed-short	int
unsigned-short	int
signed-int	int
unsigned-int	int
signed-long	int
unsigned-long	int
integer	int
DISPLAY, PICTURE contains A or X (character data)	varchar
Floating point like +9.9(5)E+99	double

Parent topic: [Conversion to or from internal data types \[page 956\]](#)

Related Information

[Attunity Streams \[page 958\]](#)
[Data Federator \[page 960\]](#)
[IBM DB2 \[page 961\]](#)
[Informix \[page 963\]](#)
[Microsoft Excel \[page 965\]](#)
[Microsoft SQL Server \[page 966\]](#)
[MySQL \[page 969\]](#)
[Netezza \[page 971\]](#)
[ODBC \[page 972\]](#)
[Oracle \[page 974\]](#)
[SQL Anywhere \[page 976\]](#)
[SAP ASE \[page 979\]](#)
[SAP Sybase IQ \[page 980\]](#)
[Teradata \[page 982\]](#)

31.3.3 Data Federator

When you import metadata from a Data Federator source into the repository, SAP Data Services converts data types.

i Note

You may change the decimal precision or scale and varchar size within the software after importing from the Data Federator data source.

Data Services conversion of Data Federator data types

Data Federator	Data Services
Decimal	Decimal precision and scale (28,6)
VarChar	VarChar (1024)

Parent topic: [Conversion to or from internal data types \[page 956\]](#)

Related Information

[Attunity Streams \[page 958\]](#)

[Cobol copybook \[page 959\]](#)

[IBM DB2 \[page 961\]](#)

[Informix \[page 963\]](#)

[Microsoft Excel \[page 965\]](#)

[Microsoft SQL Server \[page 966\]](#)

[MySQL \[page 969\]](#)

[Netezza \[page 971\]](#)

[ODBC \[page 972\]](#)

[Oracle \[page 974\]](#)

[SQL Anywhere \[page 976\]](#)

[SAP ASE \[page 979\]](#)

[SAP Sybase IQ \[page 980\]](#)

[Teradata \[page 982\]](#)

31.3.4 IBM DB2

When you import metadata from an IBM DB2 source or target into the repository, SAP Data Services converts data types.

The following table shows the conversion from IBM DB2 data types to Data Services data types.

Data Services conversion of IBM DB2 data types

DB2 data type	Converts to or from Data Services data type
bigint	int

i Note
Because int is only four bytes, data is lost during the conversion to Data Services data type.

DB2 data type	Converts to or from Data Services data type
blob	blob
	i Note This data type is not supported when importing data to column-stored template tables.
character	varchar
clob	long
	i Note This data type is not supported when importing data to column-stored template tables.
date	date
dbclob	long
	i Note This data type is not supported when importing data to column-stored template tables.
decimal	decimal
double	double
float	double
graphic	varchar
integer	int
long varchar	long
long vargraphic	long
real	real
smallint	int
time	time
timestamp	datetime
varchar	varchar
vargraphic	varchar

The following table shows the conversion from internal data types to DB2 data types in template tables or Data_Transfer transform tables.

Data Services conversion of IBM DB2 template tables and Data_Transfer transform data types

Data Services data type	DB2 data type in template table
blob	blob
date	date

Data Services data type	DB2 data type in template table
datetime	date
decimal	decimal
double	double
int	int
interval	real
long	clob
numeric	character
real	real
time	time
varchar	varchar
timestamp	date

Parent topic: [Conversion to or from internal data types \[page 956\]](#)

Related Information

[Attunity Streams \[page 958\]](#)
[Cobol copybook \[page 959\]](#)
[Data Federator \[page 960\]](#)
[Informix \[page 963\]](#)
[Microsoft Excel \[page 965\]](#)
[Microsoft SQL Server \[page 966\]](#)
[MySQL \[page 969\]](#)
[Netezza \[page 971\]](#)
[ODBC \[page 972\]](#)
[Oracle \[page 974\]](#)
[SQL Anywhere \[page 976\]](#)
[SAP ASE \[page 979\]](#)
[SAP Sybase IQ \[page 980\]](#)
[Teradata \[page 982\]](#)

31.3.5 Informix

When you import metadata from a Informix source or target into the repository, SAP Data Services converts data types.

The following table shows the conversion from Informix data types to Data Services data types.

Data Services conversion of Informix data types

Informix data type	Converts to or from Data Services data type
blob	blob
byte	blob
char	varchar
character	varchar
character varying	not supported
clob	long
date	date
datetime	datetime
dec	decimal
decimal	decimal
double	double
float	double
int	integer
integer	integer
money	decimal
numeric	decimal
real	real
serial	not supported
smallfloat	real
smallint	integer
text	long
varchar	varchar

The following table shows the conversion from internal data types to Informix data types in template tables or Data_Transfer transform tables.

Conversion from Data Services to Informix template table or Data_Transfer transform

Data Services data type	Informix data type in template table
blob	blob
date	date
datetime	datetime year to second
decimal	decimal
double	float
int	int
interval	smallfloat
long	clob

Data Services data type	Informix data type in template table
numeric	decimal
real	smallfloat
time	datetime hour to second
varchar	varchar
timestamp	datetime year to second

Parent topic: [Conversion to or from internal data types \[page 956\]](#)

Related Information

[Attunity Streams \[page 958\]](#)

[Cobol copybook \[page 959\]](#)

[Data Federator \[page 960\]](#)

[IBM DB2 \[page 961\]](#)

[Microsoft Excel \[page 965\]](#)

[Microsoft SQL Server \[page 966\]](#)

[MySQL \[page 969\]](#)

[Netezza \[page 971\]](#)

[ODBC \[page 972\]](#)

[Oracle \[page 974\]](#)

[SQL Anywhere \[page 976\]](#)

[SAP ASE \[page 979\]](#)

[SAP Sybase IQ \[page 980\]](#)

[Teradata \[page 982\]](#)

31.3.6 Microsoft Excel

When you import metadata from a Microsoft Excel source or target into the repository, SAP Data Services converts data types.

Microsoft ActiveX Data Objects (ADO) makes it possible to format and convert Excel data sources. The following table shows the conversion from ADO data types to Data Services data types when Data Services imports metadata from an Excel source or target into the repository.

Data Services conversion of Microsoft Excel ADO data types

ADO data type	Converted to Data Services data type
adDouble	double

ADO data type	Converted to Data Services data type
adCurrency	double
adBoolean	varchar
adDate	timestamp
adDBTimestamp	timestamp
ad...Char	varchar

Parent topic: [Conversion to or from internal data types \[page 956\]](#)

Related Information

[Attunity Streams \[page 958\]](#)

[Cobol copybook \[page 959\]](#)

[Data Federator \[page 960\]](#)

[IBM DB2 \[page 961\]](#)

[Informix \[page 963\]](#)

[Microsoft SQL Server \[page 966\]](#)

[MySQL \[page 969\]](#)

[Netezza \[page 971\]](#)

[ODBC \[page 972\]](#)

[Oracle \[page 974\]](#)

[SQL Anywhere \[page 976\]](#)

[SAP ASE \[page 979\]](#)

[SAP Sybase IQ \[page 980\]](#)

[Teradata \[page 982\]](#)

31.3.7 Microsoft SQL Server

When you import metadata from a Microsoft SQL Server source or target into the repository, SAP Data Services converts data types.

The following table shows the conversion from Microsoft SQL Server data types to Data Services data types.

Data Services conversion of Microsoft SQL Server data types

Microsoft SQL Server data type	Converts to or from Data Services data type
binary	not supported
bigint	decimal
bit	int

Microsoft SQL Server data type	Converts to or from Data Services data type
char	varchar
date (SQL Server 2008 and higher only)	date
datetime	datetime
datetime2 (SQL Server 2008 and higher only)	datetime
decimal	decimal
float	double
image	blob
int	int
money/smallmoney	decimal
nchar	varchar
ntext	long
numeric	decimal
nvarchar	varchar
nvarchar(max)	long
real	real
smalldatetime	datetime
smallint	int
text	long
time (SQL Server 2008 and higher only)	time
i Note On UNIX, the sub-seconds (HH:MI:SS.<ff>, where ff represents the sub-seconds) for the time data type are always set to zero (0). To get a more precise time, use the datetime or datetime2 data type.	
timestamp	not supported
tinyint	int
uniqueidentifier	varchar
varbinary	not supported
varbinary(max)	blob
varchar	varchar
varchar(max)	long
xml	long

The following table shows the conversion from internal data types to Microsoft SQL Server data types in template tables or Data_Transfer transform tables.

Data Services conversion for Microsoft SQL Server template tables or Data_Transfer transform

Data Services data type	MS SQL Server data type in template table
blob	image
date	datetime
	date (SQL Server 2008 and higher only)
datetime	datetime
	datetime2 (SQL Server 2008 and higher only)
decimal	decimal
double	float
int	int
interval	real
long	text
numeric	decimal
real	real
time	datetime
	time (SQL Server 2008 and higher only)
varchar	varchar
timestamp	datetime
	datetime2 (SQL Server 2008 and higher only)

Parent topic: [Conversion to or from internal data types \[page 956\]](#)

Related Information

[Attunity Streams \[page 958\]](#)

[Cobol copybook \[page 959\]](#)

[Data Federator \[page 960\]](#)

[IBM DB2 \[page 961\]](#)

[Informix \[page 963\]](#)

[Microsoft Excel \[page 965\]](#)

[MySQL \[page 969\]](#)

[Netezza \[page 971\]](#)

[ODBC \[page 972\]](#)

[Oracle \[page 974\]](#)

[SQL Anywhere \[page 976\]](#)

[SAP ASE \[page 979\]](#)

[SAP Sybase IQ \[page 980\]](#)

[Teradata \[page 982\]](#)

31.3.8 MySQL

When you import metadata from a MySQL source or target into the repository, SAP Data Services converts data types.

The following table shows the conversion from MySQL data types to Data Services data types.

Data Services conversion of MySQL data types

MySQL data type	Converts to or from Data Services data type
bigint	decimal
decimal	decimal
dec	decimal
bit	int
tinyint	int
bool	int
smallint	int
mediumint	int
int	int
integer	int
year	int
float	real
double	double
datetime	datetime
timestamp	datetime
date	date
time	time
varchar	varchar
nvarchar	varchar
nchar	varchar
char	varchar
enum	varchar
set	varchar
tinytext	long
text	long
mediumtext	long
longtext	long
tinyblob	blob
blob	blob

MySQL data type	Converts to or from Data Services data type
mediumblob	blob
longblob	blob

The following table shows the conversion from internal data types to MySQL data types in template tables or Data_Transfer transform tables.

Data Services conversion of MySQL template table or Data_Transfer transform

Data Services data type	MySQL data type in template table
blob	blob
date	date
datetime	timestamp
decimal	decimal
double	double
int	int
interval	float
long	text
numeric	decimal
real	float
time	time
varchar	varchar
timestamp	timestamp

Parent topic: [Conversion to or from internal data types \[page 956\]](#)

Related Information

[Attunity Streams \[page 958\]](#)

[Cobol copybook \[page 959\]](#)

[Data Federator \[page 960\]](#)

[IBM DB2 \[page 961\]](#)

[Informix \[page 963\]](#)

[Microsoft Excel \[page 965\]](#)

[Microsoft SQL Server \[page 966\]](#)

[Netezza \[page 971\]](#)

[ODBC \[page 972\]](#)

[Oracle \[page 974\]](#)

[SQL Anywhere \[page 976\]](#)

[SAP ASE \[page 979\]](#)

31.3.9 Netezza

When you import metadata from a Netezza source or target into the repository, SAP Data Services converts data types.

The following table shows the conversion from Netezza data types to Data Services data types.

Data Services conversion of Netezza data types

Netezza data type	Converts to or from Data Services data type
bigint	decimal
boolean	int
byteint	int
char	varchar
date	date
double precision	double
float	double
integer	int
interval	varchar
nchar	varchar
nvarchar	varchar
numeric	decimal
real	real
smallint	int
time	time
time with time zone	varchar
timestamp	datetime
varchar	varchar

The following table shows the conversion from internal data types to Netezza data types in template tables or Data_Transfer transform tables.

Data Services conversion of Netezza template table or Data_Transfer transform

Data Services data type	Netezza data type in template table
date	date
datetime	timestamp
decimal	numeric

Data Services data type	Netezza data type in template table
double	double precision
int	integer
interval	real
numeric	numeric
real	real
time	time
timestamp	timestamp
varchar	varchar

Parent topic: [Conversion to or from internal data types \[page 956\]](#)

Related Information

[Attunity Streams \[page 958\]](#)
[Cobol copybook \[page 959\]](#)
[Data Federator \[page 960\]](#)
[IBM DB2 \[page 961\]](#)
[Informix \[page 963\]](#)
[Microsoft Excel \[page 965\]](#)
[Microsoft SQL Server \[page 966\]](#)
[MySQL \[page 969\]](#)
[ODBC \[page 972\]](#)
[Oracle \[page 974\]](#)
[SQL Anywhere \[page 976\]](#)
[SAP ASE \[page 979\]](#)
[SAP Sybase IQ \[page 980\]](#)
[Teradata \[page 982\]](#)

31.3.10 ODBC

When you import metadata from an ODBC source or target into the repository, SAP Data Services converts data types.

The following table shows the conversion from ODBC data types to Data Services data types.

Data Services conversion of ODBC data types

ODBC data type	Converts to or from Data Services data type
bigint	decimal
char	varchar
datalink	not supported
date	date
decimal	decimal
double	double
float	double
graphic	not supported
int	int
nclob	not supported
numeric	decimal
real	real
sql_longvarchar	long
sql_longvarbinary	blob
sql_wlongvarchar	long
time	time
timestamp	datetime
tinyint	int
user-defined	not supported
varchar	varchar

The following table shows the conversion from internal data types to ODBC data types in template tables or Data_Transfer transform tables.

Data Services conversion of ODBC template table or Data_Transfer transform

Data Services data type	ODBC data type in template table
blob	sql_long varbinary
date	date
datetime	timestamp
decimal	decimal
double	double
int	int
interval	real
long	sql_long varchar
numeric	decimal
real	real

Data Services data type	ODBC data type in template table
time	time
varchar	varchar
timestamp	timestamp

Parent topic: [Conversion to or from internal data types \[page 956\]](#)

Related Information

[Attunity Streams \[page 958\]](#)
[Cobol copybook \[page 959\]](#)
[Data Federator \[page 960\]](#)
[IBM DB2 \[page 961\]](#)
[Informix \[page 963\]](#)
[Microsoft Excel \[page 965\]](#)
[Microsoft SQL Server \[page 966\]](#)
[MySQL \[page 969\]](#)
[Netezza \[page 971\]](#)
[Oracle \[page 974\]](#)
[SQL Anywhere \[page 976\]](#)
[SAP ASE \[page 979\]](#)
[SAP Sybase IQ \[page 980\]](#)
[Teradata \[page 982\]](#)

31.3.11 Oracle

When you import metadata from an Oracle source or target into the repository, SAP Data Services converts data types.

The following table shows the conversion from Oracle data types to Data Services data types.

Data Services conversion of Oracle data types

Oracle data type	Converts to or from SAP Data Services data type
char	varchar
blob	blob
clob	long
date	datetime
decimal	decimal

Oracle data type	Converts to or from SAP Data Services data type
doubleprecision	double
float	double
label	not supported
long	long
long raw	blob
nchar	varchar
nclob	long
number	int: If scale is 0 and precision is < 9 decimal: All other
nvarchar2	varchar
real	double
row	not supported
rowid	not supported
timestamp	timestamp
varchar	varchar
varchar2	varchar

The following table shows the conversion from internal data types to Oracle data types in template tables or Data_Transfer transform tables.

Data Services conversion of Oracle template table or Data_Transfer transform

Data Services data type	Oracle data type in template table
blob	blob
date	date
datetime	date
decimal	decimal
double	double
int	int
interval	real
long	clob
numeric	number
real	double
time	date
varchar	varchar2
timestamp	timestamp

Parent topic: [Conversion to or from internal data types \[page 956\]](#)

Related Information

[Attunity Streams \[page 958\]](#)

[Cobol copybook \[page 959\]](#)

[Data Federator \[page 960\]](#)

[IBM DB2 \[page 961\]](#)

[Informix \[page 963\]](#)

[Microsoft Excel \[page 965\]](#)

[Microsoft SQL Server \[page 966\]](#)

[MySQL \[page 969\]](#)

[Netezza \[page 971\]](#)

[ODBC \[page 972\]](#)

[SQL Anywhere \[page 976\]](#)

[SAP ASE \[page 979\]](#)

[SAP Sybase IQ \[page 980\]](#)

[Teradata \[page 982\]](#)

31.3.12 SQL Anywhere

When you import metadata from a SQL Anywhere source or target into the repository, SAP Data Services converts data types.

The following table shows the conversion from SQL Anywhere data types to Data Services data types.

Data Services conversion of SQL Anywhere data types

SQL Anywhere data type	Converts to or from Data Services data type
BIT	int
[UNSIGNED] TINYINT	int
[UNSIGNED] SMALLINT	int
INT or INTEGER	int
UNSIGNED INT or INTEGER	decimal(10,0)
BIGINT	decimal(19,0)
UNSIGNED BIGINT	decimal(20,0)
DEC or DECIMAL	decimal
NUMERIC	decimal
FLOAT(1-24) or FLOAT	real
FLOAT(25-53)	double

SQL Anywhere data type	Converts to or from Data Services data type
REAL	real
DOUBLE [PRECISION]	double
SMALLMONEY	decimal(10,4)
MONEY	decimal(19,4)
CHAR	varchar
NCHAR	varchar
VARCHAR	varchar
NVARCHAR	varchar
LONG VARCHAR	long
LONG NVARCHAR	long
TEXT (LONG VARCHAR)	long
NTEXT (LONG NVARCHAR)	long
XML	long
UNIQUEIDENTIFIERSTR (CHAR(26))	varchar(36)
DATE	date
TIME	time
SMALLDATETIME	datetime
DATETIME	datetime
TIMESTAMP	datetime
BINARY	blob
IMAGE	blob
LONG BINARY	blob
VARBINARY	blob
UNIQUEIDENTIFIER	blob

The following table shows the conversion from internal data types to SQL Anywhere data types in template tables or Data_Transfer transform tables.

Data Services conversion of SQL Anywhere template table or Data_Transfer transform

Data Services data type	SQL Anywhere data type in template table
blob	LONG BINARY
date	DATE
datetime	DATETIME
decimal	DECIMAL
double	DOUBLE
int	INTEGER

Data Services data type	SQL Anywhere data type in template table
interval	REAL
long	LONG VARCHAR/LONG NVARCHAR
real	REAL
time	TIME
timestamp	DATETIME
varchar	VARCHAR/NVARCHAR

i Note

When using SQL Anywhere:

- TIMESTAMP supports six-digit sub-seconds (i.e. milliseconds)
- DECIMAL supports precision up to 96 due to ACTADECIMAL_MAXPRECISION (SQL Anywhere supports precision up to 127)
- VARCHAR columns support max length of 32767 characters
- LONG columns support a maximum length of 2GB characters

Parent topic: [Conversion to or from internal data types \[page 956\]](#)

Related Information

[Attunity Streams \[page 958\]](#)

[Cobol copybook \[page 959\]](#)

[Data Federator \[page 960\]](#)

[IBM DB2 \[page 961\]](#)

[Informix \[page 963\]](#)

[Microsoft Excel \[page 965\]](#)

[Microsoft SQL Server \[page 966\]](#)

[MySQL \[page 969\]](#)

[Netezza \[page 971\]](#)

[ODBC \[page 972\]](#)

[Oracle \[page 974\]](#)

[SAP ASE \[page 979\]](#)

[SAP Sybase IQ \[page 980\]](#)

[Teradata \[page 982\]](#)

31.3.13 SAP ASE

When you import metadata from an SAP ASE source or target into the repository, SAP Data Services converts data types.

The following table shows the conversion from SAP ASE data types to Data Services data types.

Data Services conversion of SAP ASE data types

SAP ASE data type	Converts to or from Data Services data type
binary	not supported
bit	int
char	varchar
datetime	datetime
decimal	decimal
double	double
float	double
image	blob
int	int
money	decimal(20,4)
numeric	decimal
real	real
smalldatetime	datetime
smallint	int
smallmoney	decimal(12,4)
text	long
timestamp	not supported
tinyint	int
varbinary	not supported
varchar	varchar

The following table shows the conversion from internal data types to SAP ASE data types in template tables or Data_Transfer transform tables.

Data Services conversion of SAP ASE template table or Data_Transfer transform

Data Services data type	SAP ASE data type in template table
blob	image
date	datetime
datetime	datetime
decimal	decimal
double	float

Data Services data type	SAP ASE data type in template table
int	int
interval	real
long	text
numeric	decimal
real	real
time	datetime
varchar	varchar
timestamp	datetime

Parent topic: [Conversion to or from internal data types \[page 956\]](#)

Related Information

[Attunity Streams \[page 958\]](#)
[Cobol copybook \[page 959\]](#)
[Data Federator \[page 960\]](#)
[IBM DB2 \[page 961\]](#)
[Informix \[page 963\]](#)
[Microsoft Excel \[page 965\]](#)
[Microsoft SQL Server \[page 966\]](#)
[MySQL \[page 969\]](#)
[Netezza \[page 971\]](#)
[ODBC \[page 972\]](#)
[Oracle \[page 974\]](#)
[SQL Anywhere \[page 976\]](#)
[SAP Sybase IQ \[page 980\]](#)
[Teradata \[page 982\]](#)

31.3.14 SAP Sybase IQ

When you import metadata from an SAP Sybase IQ source or target into the repository, SAP Data Services converts data types.

The following table shows the conversion from SAP Sybase IQ data types to Data Services data types.

Data Services conversion of SAP Sybase IQ data types

SAP Sybase IQ data type	Converts to or from Data Services data type
bigint	decimal
binary	not supported
bit	int
blob	blob
char	varchar
clob	long
date	date
datetime	datetime
decimal	decimal
double	double
float	real
int	int
long binary	blob
long varchar	long
money	decimal(19,4)
numeric	decimal
real	real
rowid	decimal
smalldatetime	datetime
smallint	int
smallmoney	decimal(10,4)
time	time
timestamp	datetime
tinyint	int
unsigned bigint	double
unsigned int	int
varbinary	not supported
varchar	varchar

The following table shows the conversion from internal data types to SAP Sybase IQ data types in template tables or Data_Transfer transform tables.

Data Services conversion of SAP Sybase IQ template table or Data_Transfer transform

Data Services data type	SAP Sybase IQ data type in template table
blob	longbinary
date	date

Data Services data type	SAP Sybase IQ data type in template table
datetime	timestamp
decimal	decimal
double	double
int	int
interval	float
long	N/A
numeric	N/A
real	real
time	time
varchar	varchar
timestamp	timestamp

Parent topic: [Conversion to or from internal data types \[page 956\]](#)

Related Information

[Attunity Streams \[page 958\]](#)
[Cobol copybook \[page 959\]](#)
[Data Federator \[page 960\]](#)
[IBM DB2 \[page 961\]](#)
[Informix \[page 963\]](#)
[Microsoft Excel \[page 965\]](#)
[Microsoft SQL Server \[page 966\]](#)
[MySQL \[page 969\]](#)
[Netezza \[page 971\]](#)
[ODBC \[page 972\]](#)
[Oracle \[page 974\]](#)
[SQL Anywhere \[page 976\]](#)
[SAP ASE \[page 979\]](#)
[Teradata \[page 982\]](#)

31.3.15 Teradata

When you import metadata from a Teradata source or target into the repository, SAP Data Services converts data types.

The following table shows the conversion from Teradata data types to Data Services data types.

Data Services conversion of Teradata data types

Teradata data types	Converts to or from Data Services data type
bigint	decimal
blob	blob
byteint	int
clob	long
char varying (n)	varchar(n)
char [(n)]	varchar(n)
date	date
decimal	decimal
double precision	float
float	float
int	int
long varchar	long
long vargraphic	long
numeric	decimal
real	float
smallint	int
time	time
timestamp	datetime
varchar < 32000	varchar
varchar >= 32000	long
varbyte < 32000	not supported
varbyte >= 32000	blob

The following table shows the conversion from internal data types to Teradata data types in template tables or Data_Transfer transform tables.

Data Services conversion of Teradata template table or Data_Transfer transform

Data Services data type	Teradata data type in template table
blob	blob
date	date
datetime	timestamp
decimal	decimal
double	N/A
int	int
interval	N/A
long	long varchar

Data Services data type	Teradata data type in template table
numeric	N/A
real	N/A
time	time
varchar	varchar
timestamp	N/A

Parent topic: [Conversion to or from internal data types \[page 956\]](#)

Related Information

[Attunity Streams \[page 958\]](#)

[Cobol copybook \[page 959\]](#)

[Data Federator \[page 960\]](#)

[IBM DB2 \[page 961\]](#)

[Informix \[page 963\]](#)

[Microsoft Excel \[page 965\]](#)

[Microsoft SQL Server \[page 966\]](#)

[MySQL \[page 969\]](#)

[Netezza \[page 971\]](#)

[ODBC \[page 972\]](#)

[Oracle \[page 974\]](#)

[SQL Anywhere \[page 976\]](#)

[SAP ASE \[page 979\]](#)

[SAP Sybase IQ \[page 980\]](#)

31.4 Conversion of data types within expressions

When possible, SAP Data Services optimizes data flows by pushing expressions down to an underlying database manager.

In a single transaction, the software can push down expressions so that the underlying database manager performs the expression. However, when the software evaluates an expression that includes operands of more than one data type, the software attempts to convert the operands to the same data type first.

i Note

An exception to this behavior for operands of more than one data type is for national character set data types. The software can push down national character set data types when it cannot push down other expressions.

When Data Services requires that it converts a data type, it issues one of the following message types:

- If the conversion is illegal, the software issues an error. Remove the mismatch between the data types before you re execute the job.
- If the conversion is legal, the software issues a warning that indicates it will not interrupt the job execution.

i Note

When the software converts a data type to evaluate an expression, the results might not be what you expect. To avoid legal but incorrect conversions, always validate before executing and examine the circumstance of any data type conversion warnings.

Related Information

[varchar](#)

31.5 Conversion of number data types in expressions

SAP Data Services uses a type-promotion algorithm to evaluate expressions that contain more than one number data type.

Data Services ranks number data types from highest to lowest:

- decimal
- double
- real
- int

If Data Services encounters expressions that have more than one number data type among the operands, it converts all of the operands to the data type of the operand with the highest ranking type.

Example

if A is an `int` and B is a `double`, the software evaluates the expression `A+B` by first converting A to `double` and then adding the two `double` values. The result is type `double`.

If in addition to A and B, you multiply the result C, which is a `decimal`, then the software evaluates `(A+B)*C` by first converting `(A+B)` to `decimal`, and then performing the indicated operations on the two `decimal` values. The result is type `decimal`.

For addition, subtraction, and multiplication, the operation result is equal to the higher ranked of the two operands. For example:

```
int + double = double
```

31.5.1 Division algorithm

SAP Data Services uses a special algorithm when it divides number data types.

Division algorithm

Numerator data type	Denominator data type			
	int	real	double	decimal(p,s1)
int	double	double	double	decimal(p,s1)
real	double	double	double	decimal(p,s1)
double	double	double	double	decimal(p,s1)
decimal(p,s2)	decimal(p,s2)	decimal(p,s2)	decimal(p,s2)	decimal(p,max(10,s1,s2))

31.5.2 Conversion of decimals with different scale and precision

Data Services follows a specific process when it converts decimal data types that have different scale and precision settings.

If decimals of two different scales are included in a single expression, the software uses the higher of the two scales.

❖ Example

For example:

```
decimal(5,4) * decimal(7,2) = decimal(7,2)
```

When Data Services operates on two `decimals` of different scale values, it loses the precision.

❖ Example

For example, when adding a `decimal(28, 26)` to a `decimal(28,1)`, the resulting decimal value has the lower of the two scale values:

```
400000.5 + 40.00005 = 400040.5
```

The lowest scale for division that involves a decimal is 10.

31.5.3 Conversion between string and number data types

Data Services attempts to convert a string to a number data type when it encounters a string where it expects a number.

For example, in mathematical operations or functions where Data Services expects numeric arguments, Data Services attempts to convert a string to a number.

For multiplication and division operations, Data Services converts operands to numbers.

The following tables show promotion algorithms for addition, subtraction, and comparison.

Addition promotion algorithm

Provided data type	Data type required to evaluate the expression		
	Number	Date/time	String
Number	OK (promoted)	Number to interval	String to number
Date/time	Number to interval	Illegal	String to interval
String	String to number	String to interval	String to real

Subtraction promotion algorithm

Provided data type	Data type required to evaluate the expression			
	Number	Date/time	String	Interval
Number	OK (promoted)	Illegal	String to number	Interval to number
Date/time	Number to interval	OK	String to interval	OK
String	String to number	String to datetime	String to real	String to number
Interval	Number to interval	Illegal	String to interval	OK

Comparison promotion algorithm

Provided data type	Data type required to evaluate the expression			
	Number	Date/time	String	Interval
Number	OK (promoted)	Illegal	String to number	OK
Date/time	Illegal	OK	Illegal	Illegal
String	Illegal	String to datetime	OK	String to interval
Interval	OK	Illegal	String to interval	OK

31.5.4 Conversion between Oracle string and date data types

Oracle changes data to a default datetime format in certain circumstances.

If you load `datetime` data from SAP Data Services into a `char` field in an Oracle table, Oracle puts the data in its default datetime format. The default datetime format includes only date values so the conversion causes the data to lose the time value.

31.6 Conversion between explicit data types

Use function to convert data types between explicit data types.

The following table shows the built-in function to use to convert data from one type to another.

Built-in functions for converting data

Function	Description
<code>interval_to_char</code>	Converts an interval value to a string.
<code>julian_to_date</code>	Converts a Julian value to a date.
<code>num_to_interval</code>	Converts a numeric value to an interval.
<code>to_char</code>	Converts a date or numeric data type to a string.
<code>to_date</code>	Converts a string to a date based.
<code>to_decimal</code>	Converts a varchar to a decimal.

For more information about built-in functions, see [Built-in functions \[page 783\]](#).

You can also import database-specific functions to perform data type conversions.

Related Information

[interval_to_char](#)
[julian_to_date](#)
[num_to_interval](#)
[to_char](#)
[to_date](#)
[to_decimal](#)

31.7 Conversion between native data types

A Data Quality transform can get and set field data in a format other than the declared data type.

For example, if the input field is varchar, you can map it to an int data type field, as long as the varchar field contains all digits. However, SAP Data Services does not support certain conversions depending on the data type and field content.

i Note

When you map a data type to an input or output field that is an invalid data type, Data Services issues a verification error.

❖ Example

The USA Regulatory Address Cleanse transform has a varchar data type input field named `<Postcode_Full>`. Varchar field data types write to any kind of data type as long as you format the data correctly, and it contains all digits. You could map the `<Postcode_Full>` field to an integer data type because the field contains numbers. However, you could not map the `<Postcode_Full>` field to a date data type because it does not conform to the date format.

[date \[page 989\]](#)

Map date data type fields to applicable data type fields to avoid a verification error.

[datetime \[page 990\]](#)

Map datetime data type fields to applicable data type fields to avoid a verification error.

[decimal \[page 991\]](#)

Map decimal data type fields to applicable data type fields to avoid a verification error.

[double \[page 991\]](#)

Map double data type fields to applicable data type fields to avoid a verification error.

[int \(integer\) \[page 993\]](#)

Map int (integer) data type fields to applicable data type fields to avoid a verification error.

[varchar \[page 994\]](#)

Map varchar data type fields to applicable data type fields to avoid a verification error.

31.7.1 date

Map date data type fields to applicable data type fields to avoid a verification error.

Valid and invalid data types for date

date field as	Valid data type	Invalid data type
Input	date	integer
	character	double
	datetime	decimal
Output	date	integer
	character	double
	datetime	decimal

Parent topic: [Conversion between native data types \[page 988\]](#)

Related Information

[datetime \[page 990\]](#)

[decimal \[page 991\]](#)

[double \[page 991\]](#)
[int \(integer\) \[page 993\]](#)
[varchar \[page 994\]](#)

31.7.2 datetime

Map datetime data type fields to applicable data type fields to avoid a verification error.

Valid and invalid data types for datetime

datetime as	Valid data types	Invalid data types
Input	datetime character date with possible truncation time with possible truncation	integer double decimal
output	datetime character date with possible truncation time with possible truncation	integer double decimal

Parent topic: [Conversion between native data types \[page 988\]](#)

Related Information

[date \[page 989\]](#)
[decimal \[page 991\]](#)
[double \[page 991\]](#)
[int \(integer\) \[page 993\]](#)
[varchar \[page 994\]](#)

31.7.3 decimal

Map decimal data type fields to applicable data type fields to avoid a verification error.

Valid and invalid data types for decimal

decimal used as	Valid data types	Invalid data types
Input	decimal character integer with possible truncation double with possible truncation	date time datetime
Output	decimal character integer with possible truncation double with possible truncation	date time datetime

Parent topic: [Conversion between native data types \[page 988\]](#)

Related Information

[date \[page 989\]](#)

[datetime \[page 990\]](#)

[double \[page 991\]](#)

[int \(integer\) \[page 993\]](#)

[varchar \[page 994\]](#)

31.7.4 double

Map double data type fields to applicable data type fields to avoid a verification error.

Valid and invalid data types for double

double used as	Valid data types	Invalid data types
Input	double character decimal with possible truncation integer with possible truncation	date time datetime

double used as	Valid data types	Invalid data types
Output	double character decimal with possible truncation integer with possible truncation	date time datetime

Input: A transform can read from a double input field and write to the following data types:

- double
- character
- decimal (with possible truncation)
- integer (with possible truncation)

Output: A transform may write to the following data types from a double data type:

- double
- character
- decimal (with possible truncation)
- integer (with possible truncation)

Invalid input and output data types for double:

- date
- time
- datetime

Parent topic: [Conversion between native data types \[page 988\]](#)

Related Information

[date \[page 989\]](#)

[datetime \[page 990\]](#)

[decimal \[page 991\]](#)

[int \(integer\) \[page 993\]](#)

[varchar \[page 994\]](#)

31.7.5 int (integer)

Map int (integer) data type fields to applicable data type fields to avoid a verification error.

Valid and invalid data types for int (integer)

int (integer) used as	Valid data types	Invalid data types
Input	integer character decimal double	date time datetime
Output	integer character decimal double	date time datetime

Parent topic: [Conversion between native data types \[page 988\]](#)

Related Information

[date \[page 989\]](#)

[datetime \[page 990\]](#)

[decimal \[page 991\]](#)

[double \[page 991\]](#)

[varchar \[page 994\]](#)

31.7.6 varchar

Map varchar data type fields to applicable data type fields to avoid a verification error.

Valid and invalid data types for varchar

varchar used as	Valid data types	Invalid data types
Input	Any data type supported by Data Services.	The data type is invalid when you do not format the varchar data correctly. ❖ Example When you convert varchar to integer, the software does not convert data correctly when the varchar field contains characters and numbers and not exclusively numbers.
Output	Any data type supported by Data Services.	None. Data Services automatically converts field contents to varchar.

i Note

For varchar used as output, Data Services may truncate the data if the output field is not long enough.

Parent topic: [Conversion between native data types \[page 988\]](#)

Related Information

[date \[page 989\]](#)

[datetime \[page 990\]](#)

[decimal \[page 991\]](#)

[double \[page 991\]](#)

[int \(integer\) \[page 993\]](#)

32 Monitor jobs in the Administrator

The Administrator application in the SAP Data Services Management Console is the primary monitoring resource for all jobs designed in SAP Data Services Designer.

To view the overall status of a batch job and job statistics, SAP Data Services provides job monitoring tools in the Management Console Administrator. Also perform the following using the job monitoring tools:

- View job history
- Delete job history
- Stop a running job
- Delete trace monitor, and error logs

For more information about monitoring jobs, see the *Administrator Guide*.

Related Information

[Monitoring jobs](#)

33 Multi user Development

Work in a multi user development environment in SAP Data Services by using a central repository.

About multiple users

SAP Data Services supports a multi user development environment. A team can work together on an application during the development, testing, or production phase. Also, different teams can work on the different phases at the same time.

Each individual developer works on an application in their unique local repository. Each team uses a central repository to store the master copy of its application. The central repository preserves all versions of all objects in the application so users can revert to a previous version if necessary.

Related Information

[Central Object Library \[page 28\]](#)

[Multiple users \[page 998\]](#)

33.1 Central versus local repository

A central repository stores the team copy of an SAP Data Services object and acts as a source control for multi user environments.

A central repository contains all of the information that is in a local repository, such as object definitions, datastores, and so on. The central repository is a storage location for this information. To change the object, you work in a local repository.

A local repository contains copies of objects in the central repository. There are two ways to update your local repository with contents of the central repository:

- You can copy objects from the central repository to your local repository.
- You can check out objects from the central repository to your local repository.

Copying Objects

When you copy an object from the central repository, the software places the copy into your local repository if you do not already have the object. If you already have that object in your local repository, the software overwrites your local copy with the version from the central repository.

When you change your local copy:

- The object in the central repository does not change.
- You cannot upload the locally changed object to the central repository to replace the current version since you have only copied the object and not checked it out.

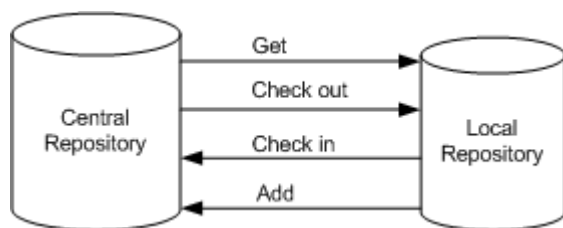
Checking Out Objects

When you check out an object from the central repository, the same things happen in your local repository that happens when you obtain a copy: either the copy is placed in your local repository if you do not already have it, or the copy you already have is overwritten with the version from the central repository. In addition, the object in the central repository becomes locked to allow you to make changes to that object that you later upload into the central repository. When you have an object checked out, no other users can check out the object. Users can copy the object, but any changes they make to their local copy remain local.

When you change your local copy:

- The object in the central repository does not change until you check your local copy into the central repository.
- After you check your changes back into the central repository, any authorized user can check out the object to change it further.
- The changes you made locally to an object and checked into the central repository are not dispersed to anyone's local copy of that object unless they then copy the object from the central repository or check out the object.

The following image shows all of the ways to access objects from the central repository and to put objects into the central repository.

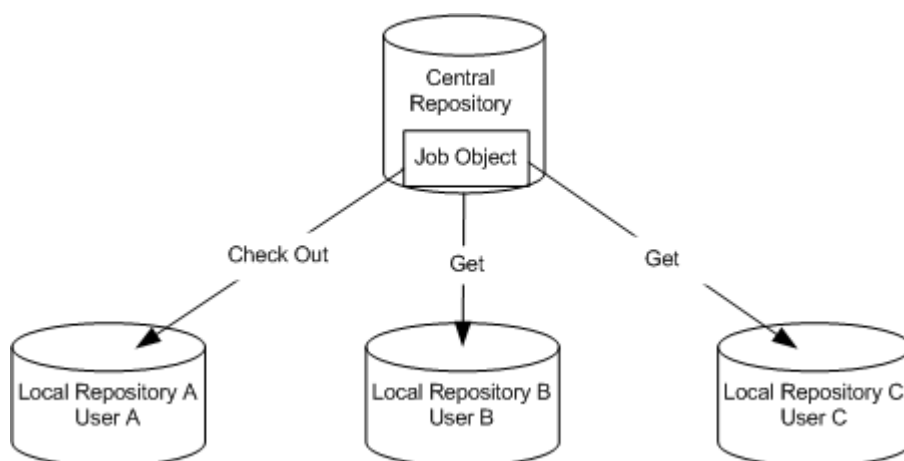


Object History

The central repository tracks all versions of an object and the dates and times the software saved the version of the object. The Central Object Library in Designer displays a brief history of an item in the [Latest Version](#) column. For more detailed information, select the [Show History](#) menu icon in the Central Object Library.

How this works

Multiple users working from unique local repositories can connect to the same central repository. These users can work on the same application and share their work. However, at any given time only one user can check out and change a particular object. While an object is checked out to one user, other users can get or copy the latest version of the object. However, other users cannot change the object in the central repository.



The central repository retains each version of an object. Version retention enables you to revert an object back to a previous version when changes that you made do not work or are wrong.

The central repository and the local repository must use the same software repository version. For example, you can run Designer version X.2 with a central and local repository version X.1. However, you cannot run Designer version X.2 with a central repository version X.1 and a local repository version X.2

Related Information

[Central Object Library \[page 28\]](#)

[Viewing and modifying permissions \[page 1009\]](#)

33.2 Multiple users

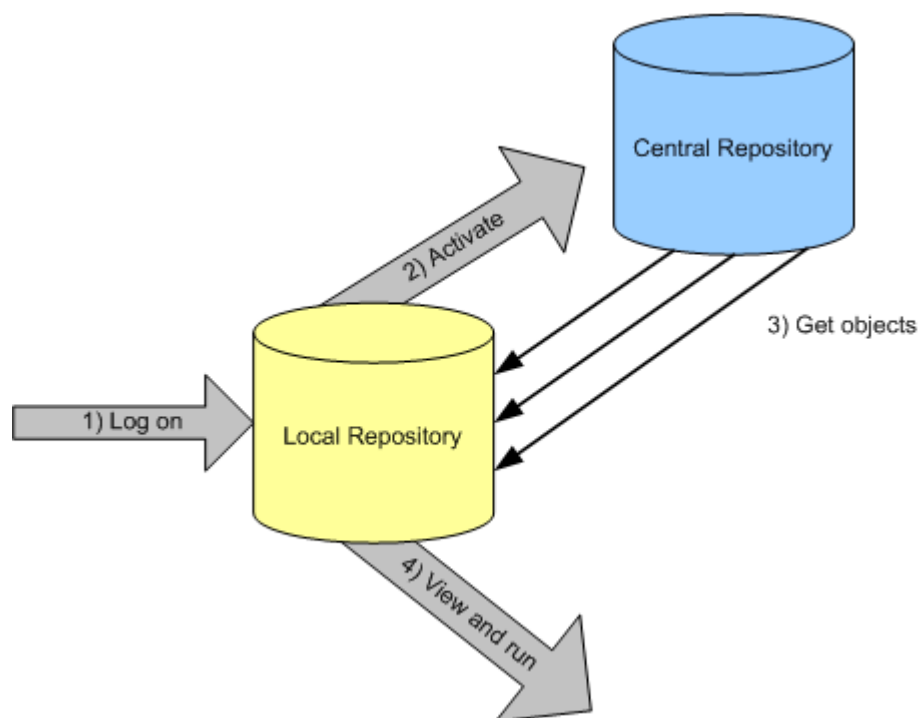
A multi-user environment affects how you use SAP Data Services and how you manage different phases of an application. For success in a multi-user environment, you must maintain consistency between your local repository and the central repository.

The following terms apply when discussing the software and multi-user environments:

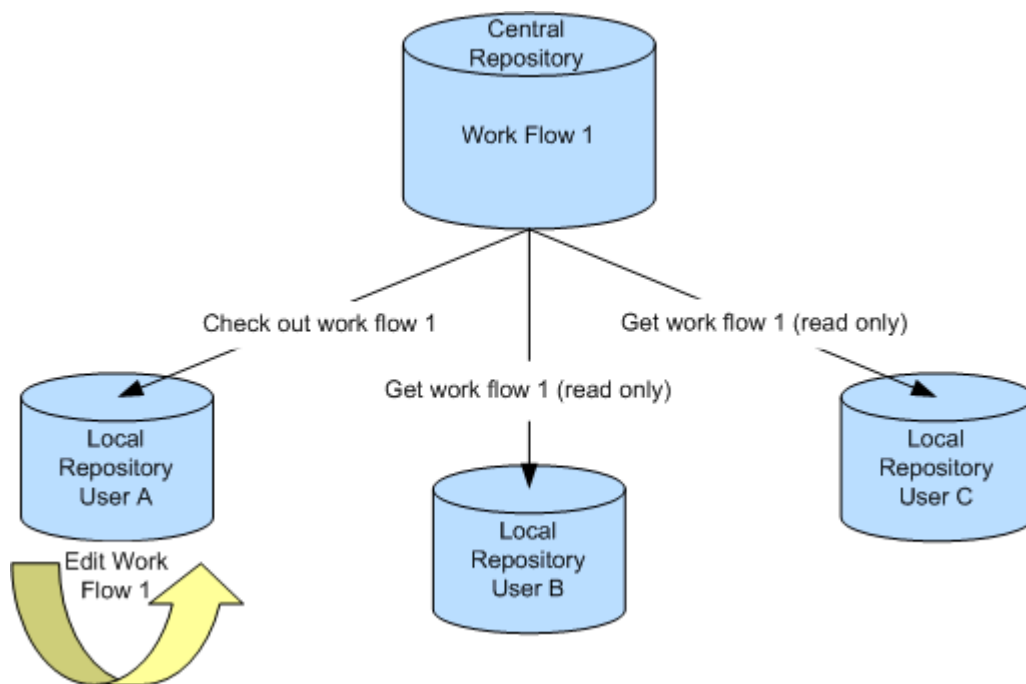
Term	Definition
Highest level object	The highest level object is the object that is not a dependent of any object in the object hierarchy. For example, if Job 1 is comprised of Work Flow 1 and Data Flow 1, then Job 1 is the highest level object.
Object dependents	Object dependents are objects associated beneath the highest level object in the hierarchy. For example, if Job 1 is comprised of Work Flow 1 which contains Data Flow 1, then both Work Flow 1 and Data Flow 1 are dependents of Job 1. Further, Data Flow 1 is a dependent of Work Flow 1.

Term	Definition
Object version	An object version is an instance of an object. Each time you add or check in an object to the central repository, the software creates a new version of the object. The latest version of an object is the last or most recent version created.

When working in a multi-user environment, you activate the link between your local repository and the corresponding central repository each time you log in. To ensure that your repository is current, you can get (copy) the latest version of each object in the central repository. Once you get an application in your local repository, you can view and run it from the Designer.



However, if you plan to make changes to objects in the application, you must check out those objects. After you check out an object, no other user can make changes. Essentially, you lock the version in the central repository; only you can change that version. Other users can only get and view the object.



When you are done making changes to an object, save those changes in the local repository and check the object back into the central repository. The software saves the changed object in the central repository and makes the object available for check-out by others. The software maintains all versions of saved objects in the central repository. Thus later, you can copy an old version of a saved object, even after replacing it in your local repository with a new version.

At any time, you can label an object or a group of objects. An object label provides a convenient mechanism for identifying objects later. For example, you may find it helpful to label objects by feature. Later, if you decide you want to eliminate a recently-added feature, you can get all objects that have the label without that feature.

You can also compare two objects—such as two different object versions in the central repository, or an object in your local repository to an object in the central repository. By comparing two objects, you can determine what parts of an object changed and decide whether you want to revert to an older version of an object.

Related Information

[Designer Guide: Design and Debug, Comparing Objects \[page 846\]](#)

33.3 Security and the central repository

You also have several options to make your central repository secure. Use these options when you need to control access and provide for object tracking within your central repository. These security options apply only to central repositories and include:

- Authentication — Allows only valid users to log in to a central repository.
- Authorization — Grants various levels of permissions to objects.
- Auditing — Maintains a history of changes made to an object including user names.

Implement security for a central repository by establishing a structure of groups and associated users using the Administrator.

Related Information

[Implementing Central Repository Security \[page 1005\]](#)

33.4 Multi-user Environment Setup

Overview of multi-user setup

To support multiple developers, configure a multi-user environment and set up several repositories. Specifically, you must:

- Create a local repository for each developer.
- Create a central repository.
- Define a connection to central repository from each local repository.
- Activate the connection to a central repository.

Related Information

[Creating a nonsecure central repository \[page 1001\]](#)

[Defining a connection to a nonsecure central repository \[page 1002\]](#)

[Activate a central repository \[page 1003\]](#)

33.4.1 Creating a nonsecure central repository

To support multiple users in a single development environment, it is recommended that you use a central repository. The central repository stores master information for the development environment.

This procedure applies to nonsecure repositories only.

1. Create a database to be used for the central repository using your database management system.
2. Choose **► Start ► Programs ► SAP Data Services 4.2 ► Data Services Repository Manager ►**.

3. In the Repository Manager window, click the [Central](#) button in the [Repository Type](#) field, and enter the database connection information for the central repository.
4. Click [Create](#).

The repository tables are created in the database you identified.

Related Information

[Implementing Central Repository Security \[page 1005\]](#)

33.4.2 Defining a connection to a nonsecure central repository

A team working on an application only needs one central repository. However, each team member requires a local repository. Furthermore, each local repository requires connection information to any central repository it must access.

This procedure applies to nonsecure repositories only.

i Note

The version of the central repository must match the version of the local repository.

1. Start the Designer and log in to your local repository.
2. Choose **Tools** > [Central Repositories](#) to open the Options window.
The Central Repository Connections option is selected in the Designer Options list.
3. Right-click in the [Central Repository Connections](#) box and select [Add](#).
The [Repository Password](#) window opens.
4. Enter the password for the central repository.
The repository appears in the [Central repository connections](#) box.
5. Click [Activate](#).
6. Again, enter the password for the central repository.
7. Click [OK](#).

Related Information

[Implementing Central Repository Security \[page 1005\]](#)

33.4.3 Activate a central repository

Before you can use your central repository, activate the link between your local repository and the central repository.

Your local repository reflects the contents of your central repository. However, the central repository may contain additional objects that are not in your local repository, or it may contain different versions of objects than what is in your local repository.

When you “get” or “check out” objects from the central repository, you copy objects from the active central repository into your local repository. Whenever you check in objects from your local repository to the central repository, you save the version from your local repository into the active central repository.

When you start Designer, always log in to a local repository. Never log into a central repository. If you do, then the central repository acts as a local repository, and you run the risk of corrupting version information. If you attempt to log in to the central repository, Data Services issues a warning. Log out immediately and log in to a local repository.

After you log in to Designer, activate the correct central repository. When you activate a central repository, the central object library opens. View a list of all objects and the check-out status of each object.

33.4.3.1 Activating a central repository

After you log in to Designer, activate the applicable central repository to view objects and other information such as check-out status.

1. Log in to Designer and the applicable local repository.
2. Choose ► **Tools** ► **Central Repositories** ► to open the **Options** dialog box.
3. In the **Central repository connections** list, select the applicable central repository.
4. If you want Designer to activate the central repository each time you log in to your local repository, check **Reactivate automatically**.
5. Right-click the central repository name and select **Activate**.

The **Options** dialog box indicates that the selected central repository is active and closes automatically. The central object library opens.

33.4.3.2 Opening the Central Object Library

Open the Central Object Library directly from SAP Data Service Designer.

Designer has an icon that makes opening the Central Object Library easy.

1. Click the **Central Object Library** icon  in the toolbar, or select ► **Tools** ► **Central Object Library** ►.

The Central Object Library looks like the Local Object Library. It shows all the objects in the repository grouped on appropriate tabs. However, the Central Object Library contains the additional columns **Check out repository** and **Latest version** for tracking objects and viewing history. Additional columns indicate group permissions and the name of the user who has checked out the object.

i Note

If the central library is secured, additional columns include [Check out user](#) and [Permission](#).

The [Central Object Library](#) dialog box appears in floating mode.

2. Drag the dialog box to the project area to dock it along with your other tools.
3. Right-click the [Central Object Library](#) tool bar and select [Floating](#) from the dropdown list to undock the dialog box.

You can also change connection information through the [Central Object Library](#) dialog box.

Related Information

[Defining a connection to a nonsecure central repository \[page 1002\]](#)

[Central Object Library \[page 28\]](#)

[Implementing Central Repository Security \[page 1005\]](#)

33.4.3.3 Changing the active central repository

Select a central repository from the list on the top of the central object library.

SAP Data Services makes the selected central repository active—objects from that repository appear in the central object library. Connection information about that repository appears in the upper right corner of the central object library.

33.4.3.4 Changing central repository connections

1. Click the [Edit Central Repository Connection](#) button on the top of the central object library.

The Options window opens with the Central Repository Connections option selected in the Designer Options list.

Alternatively, you can open the Options window by selecting **Tools** > [Central Repositories](#).

2. Select a central repository in the [Central Repository Connections](#) box, right-click, and select [Edit](#).

When the Datastore Administrator window opens:

- To disconnect from the currently active central repository, right-click the central repository in the [Central Repository Datastores](#) box and select [Deactivate](#)
- To delete connection information for a central repository, right-click the central repository in the [Central Repository Datastores](#) box and select [Delete](#).

After confirming your selection, the connection information from this local repository is deleted. You can no longer connect to that central repository from this local repository.

i Note

You are not deleting the central repository; you are only deleting the connection information between your local repository and this central repository.

- To make another repository the active central repository, right-click the central repository in the *Central Repository Datastores* box and select *Activate*.

33.5 Implementing Central Repository Security

About this section

This section describes how to implement optional security features for central repositories.

33.5.1 Overview

SAP Data Services provides options for managing secure access and tracking for objects in central repositories. Mechanisms for managing central repository security include:

- Authentication — Allows only valid users to log in to a central repository.
- Authorization — Grants various levels of permissions to objects.
- Auditing — Maintains a history of changes made to an object including user names.

Note that these security mechanisms and procedures apply only to central repositories.

33.5.1.1 Group-based permissions

You implement security for a central repository by establishing a structure of groups and associated users using the Administrator and the Central Management Console (CMC).

Access permissions for objects apply at the group level. More than one group can have the same permissions to the same object at a time. Groups are specific to a repository and are not visible in any other local or central repository.

Therefore, users do not get individual permissions. In the Designer, users select from the group(s) to which they belong, and the selected (current) group dictates their access to that object. Each user must have one default group but can belong to more than one group. When a user adds an object to a secure central repository, the user's current group automatically has Full permissions to that object.

User name and password authentication is required for every logon to a secure central repository. Users can change their passwords at any time in the CMC.

33.5.1.2 Permission levels

Each object in a secure central repository can have one of the following permissions levels:

- Full — This is the highest level of permission. The group can perform all possible actions including checking in, checking out, and deleting the object. You might assign this type of access to developers, for example.
- Read — Users can only get a copy of the object from the central repository or compare objects between their local and central object libraries. You might assign this type of access to QA, for example.
- None — Users cannot get copies of the object but can view it and its properties.

When an authenticated user adds an object to a secure central repository, the user's current group receives Full permissions to the object. All other groups receive Read permissions. Members of the group with Full permissions can change the other groups' permissions for that object.

33.5.1.3 Process summary

You implement security for a central repository by:

1. Using the Repository Manager to add a secure central repository or upgrade an existing nonsecure central repository.
2. Using the Central Management Console (CMC) to add users.
3. Using the Administrator to add the users to central repository groups.
4. Defining the connection from the Designer.
5. Adding objects to the central repository as well as view and modify object permissions.

Related Information

[Creating a secure central repository \[page 1006\]](#)

[Adding a multi-user administrator \(optional\) \[page 1007\]](#)

[Setting up groups and users \[page 1008\]](#)

[Defining a connection to a secure central repository \[page 1008\]](#)

[Working with objects in a secure central repository \[page 1009\]](#)

33.5.2 Creating a secure central repository

The first step in establishing security measures for multi-user development is to create a secure central repository or upgrade an existing nonsecure central repository.

i Note

These procedures apply to secure repositories only.

Related Information

[Multi-user Environment Setup \[page 1001\]](#)

33.5.2.1 Creating a secure central repository

1. Create a database to be used for the central repository using your database management system.
2. Choose ► [Start](#) ► [Programs](#) ► [SAP Data Services 4.2](#) ► [Data Services Repository Manager](#) ►.
3. In the Repository Manager window, click the [Central](#) button in the Repository Type field and enter the database connection information for the central repository.
4. Select the [Enable security](#) check box.
5. Click [Create](#).

The software creates repository tables in the database you identified.

33.5.2.2 Upgrading a central repository from nonsecure to secure

You can modify an existing central repository to make it secure; however, you cannot undo this change.

1. Open the Repository Manager.
2. In the Repository Manager window, click the [Central](#) button in the Repository Type field and enter the database connection information for the central repository to modify.
3. Select the [Enable security](#) check box.
4. Click [Upgrade](#).

The software updates the repository tables in the database you identified.

i Note

When you upgrade an existing non-secure central repository to a secure central repository, a new group, DIGroup, is automatically created for you and displayed in ► [Management Console](#) ► [Administrator](#) ► [Central Repositories](#) ►. To access the repository, add existing users to the group in the Administrator.

33.5.3 Adding a multi-user administrator (optional)

In the Central Management Console (CMC), you have the option of adding a user with the role of Multi-user Administrator. This role is limited to managing secure central repositories, so it is therefore a subset of the Administrator role. For example, Multi-user Administrators cannot add a local repository or a nonsecure central repository.

Multi-user Administrators can:

- Add and remove secure central repositories.
- Manage users and groups.
- View secure central repository reports.

Related Information

[Administrator Guide: User management, Group management](#)

33.5.4 Setting up groups and users

The next step in implementing central repository security is to add and configure groups and users with the Central Management Console (CMC) and the Administrator.

33.5.5 Defining a connection to a secure central repository

The next step in implementing central repository security is to define a connection to the repository in the Designer.

This procedure applies to secure central repositories only.

1. Start the Designer and log in to your local repository.
2. From the [Tools](#) menu, click [Central Repositories](#) to open the Options window.
The Central Repository Connections option should be selected in the Designer list.
3. Click [Add](#).
4. Enter your CMS connection information and click [Log On](#).
5. Select the secure central repository you want to connect.
6. Click [OK](#).

The list of central repository connections now includes the newly connected central repository and it is identified as being secure.

7. With the repository selected, click [Activate](#).
8. Click [OK](#).

Related Information

[Multi-user Environment Setup \[page 1001\]](#)

[Activate a central repository \[page 1003\]](#)

33.5.6 Working with objects in a secure central repository

Related Information

[Adding objects to the central repository \[page 1011\]](#)

[Viewing and modifying permissions \[page 1009\]](#)

33.5.6.1 Viewing and modifying permissions

After completing all configuration tasks and adding objects to the secure central repository, use the central object library to view and modify group permissions for objects.

33.5.6.1.1 Viewing permissions for an object

1. Start the Designer and log in to your local repository.
2. Open the secure central object library.

Your default group appears in the drop-down list at the top of the window and is marked with an asterisk. The Permissions column displays the current group's access level for each object. If you add a new object to the central library, the current group gets FULL permissions and all other groups get READ permission.

33.5.6.1.2 Changing object permissions to other groups

You must have Full permissions to change object access to other groups.

1. In the central object library, right-click the object and click ► *Permission* ► *CDC Adapter Configuration* ► *Object* ► or ► *Permission* ► *Object and dependants* ►.
2. The *Permission* dialog box opens, which displays a list of available groups and the group's access level for the object(s).
3. Click in the *Permission* column, and from the drop-down list select a permission level for the group.
4. Click *Apply* or *OK*.

33.5.6.1.3 Changing the current group or the default group

1. To change the current group, in the central object library select a group from the drop-down box.
2. To change your default group, select the desired group from the drop-down box and click the save icon.
The software marks the default group with an asterisk.

33.6 Working in a Multi-user Environment

To obtain optimal results from development in a multi-user environment, it is recommended certain processes, such as checking in and checking out objects that you change, and establishing a set of conventions that your team follows, such as labeling objects.

[Filtering \[page 1010\]](#)

[Adding objects to the central repository \[page 1011\]](#)

[Checking out objects \[page 1013\]](#)

[Undoing check out \[page 1016\]](#)

[Checking in objects \[page 1017\]](#)

[Labeling objects \[page 1019\]](#)

[Getting objects \[page 1022\]](#)

[Comparing objects \[page 1023\]](#)

[Viewing object history \[page 1024\]](#)

A main feature of the central repository is that it retains object history for information and reporting.

[Deleting objects \[page 1026\]](#)

33.6.1 Filtering

SAP Data Services allows you to customize by filtering (selectively changing) environment-specific information in object definitions. Application objects can contain repository-specific information. For example, datastores and database tables might refer to a particular database connection unique to a user or a phase of development. When multiple users work on an application, they can change repository-specific information.

Specifically, filtering allows you to:

- Change datastore and database connection information
- Change the root directory for files associated with a particular file format
- Select or clear specific dependent objects

The filtering process is available when adding, checking in, checking out, or getting labeled or latest objects in a central repository.

When you select any command that uses the filtering option:

1. The *Version Control Confirmation* window displays your selected object and any dependent objects. You can exclude objects by selecting the object and changing the *Target status* from *create* to *exclude*.
2. The *Datastore Options* window shows any datastores used by the object. This window only opens if the objects that you are adding, checking in, or checking out include a datastore.

Parent topic: [Working in a Multi-user Environment \[page 1010\]](#)

Related Information

[Adding objects to the central repository \[page 1011\]](#)

[Checking out objects \[page 1013\]](#)

[Undoing check out \[page 1016\]](#)

[Checking in objects \[page 1017\]](#)

[Labeling objects \[page 1019\]](#)

[Getting objects \[page 1022\]](#)

[Comparing objects \[page 1023\]](#)

[Viewing object history \[page 1024\]](#)

[Deleting objects \[page 1026\]](#)

33.6.2 Adding objects to the central repository

After creating a central repository, connecting it to the local repository, and activating the central repository, you can add objects from the local repository to the central repository. Remember that you do all design work—the creation of jobs, work flows, and data flows—in a local repository. Therefore, you use a local repository for the initial creation of any objects in an application. After the initial creation of an object, you add it to the central repository. Once in the central repository, the object is subject to version control and can be shared among users.

You can add a single object to the central repository, or you can add an object with all of its dependents to the central repository. When you add a single object, such as a data flow, you add only that object. No dependent objects are added.

You can add objects to the central repository at any point. However, you cannot add an object that already exists in the central repository.

You cannot add a read-only transform configuration to the repository. You can, however, replicate a transform configuration and add the replica to the repository.

Parent topic: [Working in a Multi-user Environment \[page 1010\]](#)

Related Information

[Filtering \[page 1010\]](#)

[Checking out objects \[page 1013\]](#)

[Undoing check out \[page 1016\]](#)

[Checking in objects \[page 1017\]](#)

[Labeling objects \[page 1019\]](#)

[Getting objects \[page 1022\]](#)

[Comparing objects \[page 1023\]](#)

[Viewing object history \[page 1024\]](#)

33.6.2.1 Adding a single object to the central repository

1. Open the local object library.
2. Right-click the object and select ► *Add to Central Repository* ► *Object* ►.
3. The Comments window opens. Enter any comments in the *Comments* field, and click *OK*.

The software adds the object to the active central repository.

i Note

The Add to Central Repository command is not available if the object already exists in the central repository.

33.6.2.2 Adding an object and its dependent objects to the central repository

1. Open the local object library.
2. Right-click the object and select either ► *Add to Central Repository* ► *Object and dependents* ► or ► *Add to Central Repository* ► *With filtering* ► (if filtering is required).
3. The Comments window opens. Enter any comments in the *Comments* field, and click *OK*.
4. If you selected *With filtering*, complete the filtering windows.
5. Click *Finish* to add the selected objects.

Alternatively, you can select the object and drag it to the central object library to add the object and its dependents to the central repository. The filtering windows are displayed.

i Note

The *Add to Central Repository* command is not available if the object already exists in the central repository. However, the *Add to Central Repository* command is available if the object's dependents already exist in the central repository but the object itself does not.

i Note

You cannot add a read-only transform configuration to the repository. To do so, you must create a new repository, upgrade the existing repository, or import ATL that contains a new version of read-only transform configurations.

Related Information

[Filtering \[page 1010\]](#)

33.6.3 Checking out objects

When you might change any of the objects in an application, you should check out the objects that you expect to change. When you check out an object, you make that object unavailable to other users—other users can view the object but cannot make changes to the object. Checking out an object ensures that two users do not make conflicting changes to the object simultaneously.

Data Services changes the object icons in both the local and central object libraries to indicate that the object is checked out.

When an object is checked out, your central object library shows you the local repository that has checked out the object. Based on the repository name, you can determine which user is working with that object.

To see periodic changes, refresh the central object library by clicking on the [Refresh Central Object Library](#) button in the toolbar of the central object library.

Choose a check-out command based on what you will do to an object.

Parent topic: [Working in a Multi-user Environment \[page 1010\]](#)

Related Information

[Filtering \[page 1010\]](#)

[Adding objects to the central repository \[page 1011\]](#)

[Undoing check out \[page 1016\]](#)

[Checking in objects \[page 1017\]](#)

[Labeling objects \[page 1019\]](#)

[Getting objects \[page 1022\]](#)

[Comparing objects \[page 1023\]](#)

[Viewing object history \[page 1024\]](#)

[Deleting objects \[page 1026\]](#)

33.6.3.1 Checking out single objects or objects with dependents

Dependents are objects used by another object—for example, data flows that are called from within a work flow. You can check out a single object or an object with all of its dependents (as calculated in the central repository). For example, you can simply check out a work flow. In that case, you can change that work flow,

such as adding a new script to the work flow; however, you cannot change dependent objects in the work flow, such as data flows, and retain the changes in the central repository. Changes to dependent objects will only be retained in the local repository. Alternatively, you can check out the work flow with all of its dependents. In that case, you can make changes to the work flow or any of its dependents and retain the changes in both central and local repositories.

Generally, it is safest to check out an object with all dependents. When you do this, you prevent others from accidentally changing dependent objects.

33.6.3.1.1 Checking out a single object

1. Open the central object library.
2. Right-click the object you want to check out.
3. Choose ► [Check Out](#) ► [Object](#) ▾.

Alternatively, you can select the object in the central object library, and click the [Check Out Object](#) button on the top of the central object library.

The software copies the most recent version of the selected object from the central repository to your local repository, then marks the object as checked out.

33.6.3.1.2 Checking out an object and its dependent objects

1. Open the central object library.
2. Right-click the object you want to check out.
3. Choose ► [Check Out](#) ► [Object and dependents](#) ▾.

Alternatively, you can select the object in the central object library, and click the [Check Out Object and dependents](#) button on the top of the central object library.

SAP Data Services copies the most recent version of the selected object and all of its dependent objects from the central repository and marks these objects as checked out.

If a dependent object is checked out by you or another user, then the software alerts you with a Check Out Alert window, asking to get the latest version of the checked out object.

Related Information

[Getting objects \[page 1022\]](#)

33.6.3.2 Check out single objects or objects with dependents without replacement

When you check out an object, you can replace the object in your local repository with the latest version from the central repository, or you can leave the current version in your local repository intact.

When you check out an object, SAP Data Services copies the object definition from the central repository and replaces any existing definitions for that object in your local repository.

You can check out objects without replacing the objects in your local repository. For example, suppose you are working in your local repository and you make a change to an object that is not checked out. If you determine that the change improves the design or performance of your application, you will want to include that change in the central repository.

To do this, check out the object without replacing the object in your local repository—the object that you have already improved with a change. Then, check the changed object back into the central repository.

i Note

Use caution when checking out objects without replacing the version in your local repository. When you do not replace the version in your local repository, you can lose changes that others have incorporated into those objects.

33.6.3.2.1 Checking out an object or an object and its dependent objects without replacement

1. Open the central object library.
2. Right-click the object you want to check out and choose ► *Check Out* ► *Object* ► *without replacement* ► to check out the single object or choose ► *Check Out* ► *Object and dependents without replacement* ► to check out the object and all of its dependent objects.

SAP Data Services marks all appropriate objects as checked out—in both the object library and in the workspace—but does not copy any objects from the central repository to the local repository.

33.6.3.3 Check out objects with filtering

When you check out an object with filtering, the object and all its dependents are checked out.

i Note

When you check out objects with filtering, you always replace local versions with the filtered objects from the central repository.

33.6.3.3.1 Checking out an object and its dependent objects with filtering

1. Open the central object library.
2. Right-click the object you want to check out and choose ► *Check Out* ► *With filtering* ▾.
3. Complete the filtering windows.
4. Click *Finish* to check out the selected objects.

Related Information

[Filtering \[page 1010\]](#)

33.6.4 Undoing check out

Occasionally, you may decide that you did not need to check out an object because you made no changes. Or, you may decide that the changes you made to a checked-out object are not useful and you prefer to leave the master copy of the object as is. In these cases, you can undo the check out.

When you undo a check out:

- the object in the central repository remains as it was before the checkout; no changes are made and no additional version is saved in the central repository. Only the object status changes from checked out to available.
- the local version of the object maintains the changes you made. If you want the local object to be an exact copy of the central object, perform a *Get latest* operation on that object.

After you undo a check out, other users can check out and make changes to the object.

Parent topic: [Working in a Multi-user Environment \[page 1010\]](#)

Related Information

[Filtering \[page 1010\]](#)

[Adding objects to the central repository \[page 1011\]](#)

[Checking out objects \[page 1013\]](#)

[Checking in objects \[page 1017\]](#)

[Labeling objects \[page 1019\]](#)

[Getting objects \[page 1022\]](#)

[Comparing objects \[page 1023\]](#)

[Viewing object history \[page 1024\]](#)

[Deleting objects \[page 1026\]](#)

33.6.4.1 Undoing single object check out

1. Open the central object library.
2. Select a checked-out object.
3. Click the [Undo object check out](#) button. Alternatively, right-click the object and select ► [Undo Check Out](#) ► [Object](#) ▾.

SAP Data Services removes the check-out symbol and makes no changes to the object in the central repository. Any checked-out dependent objects remain checked out.

33.6.4.2 Undoing check out of an object and its dependents

1. Open the central object library.
2. Select the checked-out object that is the highest level for which you want to undo the check out.
3. Click the [Undo object and dependents check out](#) button. Alternatively, you can right-click the object in the central object library and select ► [Undo Check Out](#) ► [Object and dependents](#) ▾.

SAP Data Services removes the check-out symbols for the object and any dependent objects that are also checked out. No changes are made to these objects in the central repository.

33.6.5 Checking in objects

After you finish making changes to checked out objects, you must check them back into the central repository. Checking in objects creates a new version in the central repository, and allows others to get the changes that you have made. Checking in objects also preserves a copy of the changes for revision control purposes. Later, you can get a particular version of a checked in object and compare it to subsequent changes or even revert to the previous version.

Check in an object when you are done making changes, when others need the object that contains your changes, or when you want to preserve a copy of the object in its present state.

Choose a check-in command based on what you will do to an object.

Parent topic: [Working in a Multi-user Environment \[page 1010\]](#)

Related Information

[Filtering \[page 1010\]](#)

[Adding objects to the central repository \[page 1011\]](#)

[Checking out objects \[page 1013\]](#)
[Undoing check out \[page 1016\]](#)
[Labeling objects \[page 1019\]](#)
[Getting objects \[page 1022\]](#)
[Comparing objects \[page 1023\]](#)
[Viewing object history \[page 1024\]](#)
[Deleting objects \[page 1026\]](#)
[Checking in single objects, objects with dependents \[page 1018\]](#)
[Checking in an object with filtering \[page 1019\]](#)

33.6.5.1 Checking in single objects, objects with dependents

Just as you can check out a single object or an object with all dependent objects, you can check in a single object or an object with all checked-out dependent objects (as calculated in the local repository).

33.6.5.1.1 Checking in a single object

1. Open the central object library.
2. Select the object you want to check in.
3. Click [Check in object](#) button at the top of the central object library.

Alternatively, you can right-click the object in the central object library and select  [Check In](#)  [Object](#) .

4. A Check In window opens with a [Comment](#) box, in which you can enter comments. After entering any comments, click [OK](#).

SAP Data Services copies the object from your local repository to the central repository, and removes the check-out mark.

33.6.5.1.2 Checking in an object and its dependent objects

1. Open the central object library.
2. Select the highest level object you want to check in.
3. Click [Check in object and dependents](#) button at the top of the central object library.

Alternatively, you can right-click the object in the central object library and select  [Check In](#)  [Object](#)  and dependents.

4. A Check In window opens with a [Comment](#) box, in which you can enter comments. After entering any comments, click [OK](#).

SAP Data Services copies the selected object and all of its dependent objects from your repository to the central repository and removes the check-out mark.

33.6.5.2 Checking in an object with filtering

Just as you could check out objects with filtering, you can check in objects with filtering. When you check in an object with filtering, the object and all its dependent objects are checked in.

33.6.5.2.1 Checking in an object with filtering

1. Open the central object library.
2. Right-click the object you want to check out and choose ► *Check In* ► *With filtering* ▾.
3. A Check In window opens with a *Comment* box, in which you can enter comments. After entering any comments, click *OK*.

SAP Data Services warns you that you are about to create a new version of the object in the central repository.

4. Click *Yes* to continue with the check in.
5. Complete the filtering windows.
6. Click *Finish* to check in the selected objects.

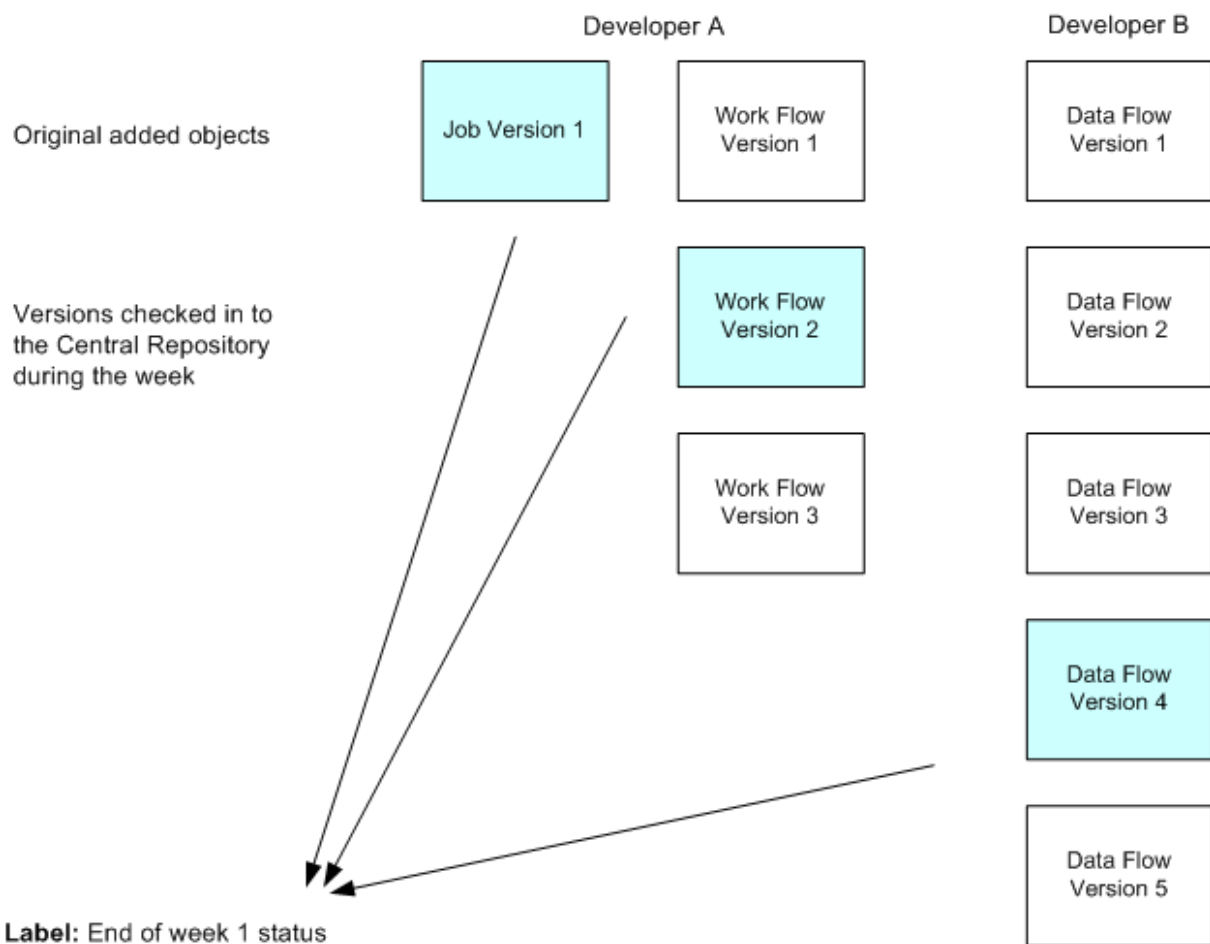
Related Information

[Filtering \[page 1010\]](#)

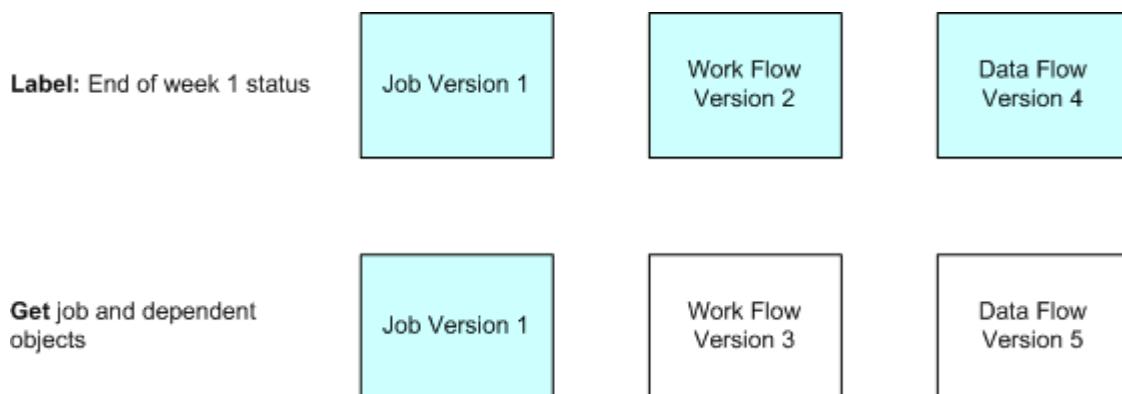
33.6.6 Labeling objects

To help organize and track the status of objects in your application, you can label objects. You can choose to either label an object, or label an object and all of its dependent objects. A label not only describes an object, but also allows you to maintain relationships between various versions of objects.

For example, suppose developer A adds a job to the central repository and works on a work flow in that job while developer B works on a data flow in the same job. At the end of the week, after developer A checks in two versions of the work flow and developer B checks in four versions of the data flow to the central repository, the job is labeled "End of week 1 status." This label contains version 1 of the job, version 2 of the work flow, and version 4 of the data flow. Both developers can continue to change their respective work flow and data flow.



At some later point, if you want to get the job with the version of the data flow with this label, getting the job by its label accomplishes this, whereas checking out the job and its dependents does not.



The label "End of week 1 status" serves the purpose of collecting the versions of the work flow and data flow that were checked in at the end of the week. Without this label, you would have to get a particular version of each object in order to reassemble the collection of objects labeled "End of week 1 status."

Parent topic: [Working in a Multi-user Environment \[page 1010\]](#)

Related Information

[Filtering \[page 1010\]](#)

[Adding objects to the central repository \[page 1011\]](#)

[Checking out objects \[page 1013\]](#)

[Undoing check out \[page 1016\]](#)

[Checking in objects \[page 1017\]](#)

[Getting objects \[page 1022\]](#)

[Comparing objects \[page 1023\]](#)

[Viewing object history \[page 1024\]](#)

[Deleting objects \[page 1026\]](#)

[Getting objects \[page 1022\]](#)

33.6.6.1 Labeling an object and its dependents

1. Open the central object library.
2. Right-click the object you want to label and choose  *Label Latest Version*  *Object*  to label only the highlighted object, or choose *Object and dependents* to label the highlighted object and all its related objects.

The *Label Latest Version* window opens.

3. In the *Label* box, enter text that describes the current status of the object, then click *OK*.

The label is inserted in the history of the object and its dependents.

Related Information

[Viewing object history \[page 1024\]](#)

33.6.6.2 Getting a labeled object with filtering

The filtering option for the Get by label operation allows you to filter (selectively change) environment-specific information in object definitions when working in a multi-user environment.

1. Open the central object library.
2. Select the highest level object you want to get.

3. Right-click the object in the central object library and select ► *Get By Label* ► *With filtering* ▾.
4. Complete the filtering window.
5. Click *Finish* to get the selected objects.

Related Information

[Filtering \[page 1010\]](#)

33.6.7 Getting objects

To make sure that your repository is up-to-date, you "get" objects. When you get an object, you copy the latest version of that object in the central object library and copy it into your local repository, replacing the version in your local repository. When you get an object, you do not check out the object. The object remains free for others to check out and change.

You can get an object with or without dependent objects and filtering.

Parent topic: [Working in a Multi-user Environment \[page 1010\]](#)

Related Information

[Filtering \[page 1010\]](#)

[Adding objects to the central repository \[page 1011\]](#)

[Checking out objects \[page 1013\]](#)

[Undoing check out \[page 1016\]](#)

[Checking in objects \[page 1017\]](#)

[Labeling objects \[page 1019\]](#)

[Comparing objects \[page 1023\]](#)

[Viewing object history \[page 1024\]](#)

[Deleting objects \[page 1026\]](#)

[Viewing object history \[page 1024\]](#)

33.6.7.1 Getting a single object

1. Open the central object library.
2. Select the object you want to get.
3. Click *Get latest version of object* at the top of the central object library.

Alternatively, right-click the object in the central object library and select ► [Get Latest Version](#) ► [Object](#) ►.

The most recent version of the object in the central repository is copied to your local repository.

33.6.7.2 Getting an object and its dependent objects

1. Open the central object library.
2. Select the highest level object you want to get.
3. Click [Get latest version of objects and dependents](#) at the top of the central object library.

Alternatively, right-click the object in the central object library and select ► [Get Latest Version](#) ► [Object and dependents](#) ►.

The most recent version of the selected object and all dependent objects from the central repository is copied to your local repository.

33.6.7.3 Getting an object and its dependent objects with filtering

1. Open the central object library.
2. Select the highest level object you want to get.
3. Right-click the object in the central object library and select ► [Get Latest Version](#) ► [With filtering](#) ►.
4. Complete the filtering windows.
5. Click [Finish](#) to get the selected objects.

Related Information

[Filtering \[page 1010\]](#)

33.6.8 Comparing objects

SAP Data Services allows you to compare two objects from local and central repositories to determine the differences between those objects.

Parent topic: [Working in a Multi-user Environment \[page 1010\]](#)

Related Information

[Filtering \[page 1010\]](#)

[Adding objects to the central repository \[page 1011\]](#)

[Checking out objects \[page 1013\]](#)

[Undoing check out \[page 1016\]](#)

[Checking in objects \[page 1017\]](#)

[Labeling objects \[page 1019\]](#)

[Getting objects \[page 1022\]](#)

[Viewing object history \[page 1024\]](#)

[Deleting objects \[page 1026\]](#)

33.6.9 Viewing object history

A main feature of the central repository is that it retains object history for information and reporting.

The central repository retains a history of all changes made to objects in the central repository. Use this history to help manage and control development of your application.

When you view objects in the Central Object Library in Designer, you can quickly view version, date, and time stamp information. View more detailed information by selecting the [Show History](#) icon.

Parent topic: [Working in a Multi-user Environment \[page 1010\]](#)

Related Information

[Filtering \[page 1010\]](#)

[Adding objects to the central repository \[page 1011\]](#)

[Checking out objects \[page 1013\]](#)

[Undoing check out \[page 1016\]](#)

[Checking in objects \[page 1017\]](#)

[Labeling objects \[page 1019\]](#)

[Getting objects \[page 1022\]](#)

[Comparing objects \[page 1023\]](#)

[Deleting objects \[page 1026\]](#)

33.6.9.1 Examining the history of an object

Users and administrators can monitor the changes made to each version of an object by looking at the object history.

1. Select **Tools** > **Central Object Library** in SAP Data Services Designer.

The Central Object Library opens in floating mode.

2. Click an object to select it.

Scan the object information in the Central Object Library to learn the following:

- Whether or not an object is checked out. An object is checked out when a red checkmark appears next to the object name. An object is available to check out when no red checkmark appears next to the object name.
- Location for the checked out object. When an object is checked out, the name of the local repository to which the object is checked out.
- Latest change information: For each object listed, the current version number, the date and time the object was last checked into the central repository, and a description of the last change.

3. Click the **Show History** button at the top of the Central Object Library.

Alternatively, right-click the object in the Central Object Library, and choose **Show History**. The **History** dialog box shows information about each revision of the object.

The following table compares the information available in the history page and in the Central Object Library.

Column	History page	Central Object Library
<i>Version</i>	The object revision number. Each time a user saves the object, the software creates a new version.	Same
<i>Label</i>	Text that a user enters to describe the status of the object at a given point.	N/A
<i>Repository</i>	Information about the local repository from which the software saved this version of the object and the user-name.	If the object is currently checked out, the name of the local repository to which the object is checked out.
<i>Date</i>	The date and time the software saved this version of the object.	Same
<i>Action</i>	The type of change a user made to the object. This table records actions such as: Checked in — User checked in object	N/A

Column	History page	Central Object Library
Comment	Comments a user enters when adding an object or checking it into a central repository.	N/A
Description	N/A	Description of the object entered when the object was created.

When the central repository is secured, users can see the user name of the person who has the object checked out, and that user's permission.

33.6.9.2 Getting a previous version of an object

1. Select an object.
2. Click the [Show History](#) button at the top of the central object library.
3. Click the version of the object you want.
4. Click the [Get Obj By Version](#) button.

i Note

When you get a previous version of an object, you only get the object but not its dependent objects.

33.6.9.3 Getting an object with a particular label

1. Select an object.
2. Click the [Show History](#) button at the top of the central object library.
3. Click the version of the object with the particular label you want.
4. Click the [Get By Label](#) button.

33.6.10 Deleting objects

You can delete objects from either the central repository or a local repository. To delete an object from the central repository, right-click the object in the central object library and select [Delete](#). To delete an object from the local repository, right-click on the object in the object library and select [Delete](#).

When you delete an object from a local repository, you do not automatically delete that object from the active central repository. In fact, you can get the object from the central repository to re-insert it.

Similarly, when you delete an object from a central repository, you do not automatically delete the object from connected local repositories. Until you delete the object from the local repository, you can add the object back to the central repository.

When you delete objects from the central repository, you only delete the selected object and all versions of the selected object; you do not delete any dependent objects.

Parent topic: [Working in a Multi-user Environment \[page 1010\]](#)

Related Information

[Filtering \[page 1010\]](#)

[Adding objects to the central repository \[page 1011\]](#)

[Checking out objects \[page 1013\]](#)

[Undoing check out \[page 1016\]](#)

[Checking in objects \[page 1017\]](#)

[Labeling objects \[page 1019\]](#)

[Getting objects \[page 1022\]](#)

[Comparing objects \[page 1023\]](#)

[Viewing object history \[page 1024\]](#)

33.7 Migrating Multi-user Jobs

Overview of multi-user job migration

Job migration applies to SAP Data Services on multiple levels: application level, repository management level, and product upgrade level. Application migration is much more flexible in a multi-user environment, allowing you to maintain not only multiple versions of your objects during development, but also during test and production phases if you choose.

Related Information

[Application phase management \[page 1027\]](#)

[Copying contents between central repositories \[page 1029\]](#)

[Central repository migration \[page 1030\]](#)

33.7.1 Application phase management

Typically, applications pass through different phases on the way from development to production. For example, an application might pass through three phases:

- Developers creating an application
- Testers validating the application
- Administrators running the application

A single central repository can support your application through all phases. Use job labeling and projects to maintain application components independently for each phase. For example, if development wants to make a certain version of an application ready for testing, they may label it "APPL_V1". Testers can then get that particular application version using the label and proceed with testing. If testing is successful, an administrator can get the application to run in the production environment. In addition, datastore configurations and file locations allows you to configure the application to run in each local environment.

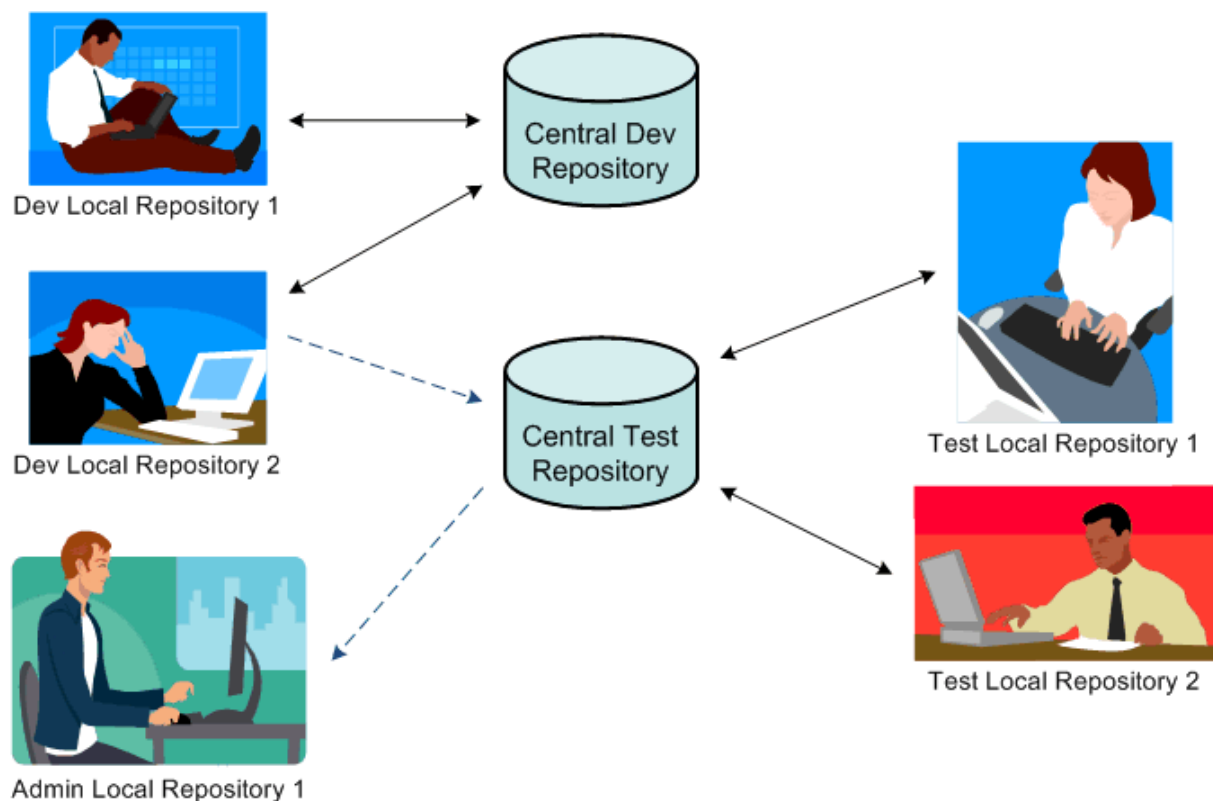
In some situations, you may require more than one central repository for application phase management. If you choose to support multiple central repositories, use a single local repository as a staging location for the transition.

In some situations, you may require more than one central repository for application phase management. Following the example above, once developers create an application version ready for testing by labeling it, a tester would get that version from the development central repository, test it and then check it into a test central repository.

That test central repository will contain all versions tested over time, allowing flexibility for testers to go back to any previous version without relying on the development environment. When an application version passes testing, an administrator can get it from the test repository and make it available in production. Again, if you need to maintain previous versions of an application already in production, you can create another central repository.

With this scheme, a developer will never interfere with the test environment, and a tester will never interfere with a production environment, creating an extremely safe process of migration.

Note that if you choose to support multiple central repositories, use a single local repository as a staging location for file transition.



33.7.2 Copying contents between central repositories

You cannot directly copy the contents of one central repository to another central repository. Rather, you must use your local repository as an intermediate repository.

33.7.2.1 Copying the contents of one central repository to another central repository

1. Activate the central repository whose contents you will copy.
2. Get the latest version of all objects in this active central repository so they exist in your local repository.
3. Activate the central repository into which you want to copy the contents.
4. The first time you copy the contents, add the objects from your local repository into this central repository.

However, if you must re-copy the contents of one central repository into another (for example, during your testing phase some part of a job was reassigned to the development phase for redesign), the process is slightly more complex:

- a. First check out specific objects without replacement from the second central repository.
- b. From your local repository, get the latest version of the objects from the first (for example, development) central repository.

- c. Then, instead of adding, check in the updated objects from your local repository to the second (for example, test) central repository.

Related Information

[Activate a central repository \[page 1003\]](#)

[Getting objects \[page 1022\]](#)

[Adding objects to the central repository \[page 1011\]](#)

[Checking out objects \[page 1013\]](#)

[Checking in objects \[page 1017\]](#)

33.7.3 Central repository migration

When you upgrade your version of SAP Data Services, you should migrate your central repository to the new version. It is recommended that you consider the following guidelines when migrating a central repository to a new release of the software.

1. Back up all central repository (as well as local repository) database tables and associated data before upgrading.
2. Maintain a separate central repository for each version of SAP Data Services to preserve object history.
To preserve the current version and history of objects in your central repository, create a new central repository of your current version of the software and copy the contents of the original central repository to the newly-created one. This way, the second central repository acts as a backup for your objects and associated history information from the older version of the software.
When you install the new version of the software, upgrade the newly-created central repository to the latest version of the software.
3. Coordinate efforts to upgrade your central repositories and local repositories at the same time.
Different versions of your central and local repository may not work together. You cannot perform a multi-user operation between a local and central repository of a different software version.
4. Check in all objects (or undo check-outs if objects were not modified after they were checked out) before migrating the central repositories.
If you cannot upgrade your central and local repositories at the same time, you should check in all objects (or undo check-outs if objects were not modified during check-out), especially those objects checked out to a local repository you will not be immediately upgrading. After you upgrade your central repository to the new version, you will not be able to check in objects from the local repository of the older version of the software.

Related Information

[Copying contents between central repositories \[page 1029\]](#)

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon  : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
 - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
 - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon  : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.

© 2021 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.