PUBLIC

SAP Data Intelligence

2022-08-06

ABAP Integration Guide

© 2022 SAP SE or an SAP affiliate company. All rights reserved.

THE BEST RUN

Content

1 User Guide for ABAP Integration in SAP Data Intelligence

This guide is relevant for you if your use case for SAP Data Intelligence Cloud involves ABAP-based SAP systems, such as SAP S/4HANA or SAP Business Information Warehouse (BW) - directional as well as bi-directional system interaction.

It focuses on the specific aspects for these use cases and does not replace the existing information for SAP Data Intelligence Cloud.

Related Information

Installation-Related Information [page 3]
Security Information [page 4]
Useful SAP Notes and Blogs [page 5]
Use Cases [page 6]
Data Extraction [page 7]
Operators [page 13]
Troubleshooting [page 17]

1.1 Installation-Related Information

What exactly you need to do to install and set up the ABAP integration features depends on the system landscape and release levels you want to work with:

- If you want to use an SAP S/4HANA system, consider the relevant release-specific SAP Note:
  - SAP S/4HANA 1909: SAP Note 2830276
  - SAP S/4HANA 2020: SAP Note 2943599
  - SAP S/4HANA 2021: SAP Note 3085579
- For older SAP systems (in combination with the DMIS add-on), consider the information provided in the relevant SAP Notes for your DMIS release, for example:
  - DMIS 2011 SP21: SAP Note 3095800
  - DMIS 2018 SP06: SAP Note 3085541
  - DMIS 2020 SP02: SAP Note 3085541
- If you want to integrate older SAP systems or SAP Busines Warehouse (BW) systems using ODP (in combination with the DMIS add-on), consider the information provided in SAP Notes 2890171 and 2775549.

> i Note
>
> Whenever possible, install and use the latest versions of the relevant software for your use case.

Working With TCIs

If you are using an older SAP S/4 HANA release and an upgrade is not an option for you, please implement the relevant transport-based correction instruction (TCI) for your release level (as listed below) to get important new functionality and security features. For example, you can then use generation 2 operators.

- SAP S/4 HANA 1909: TCI 3105890
- SAP S/4 HANA 2020: TCI 3105880
- SAP S/4 HANA 2021: TCI 3115128

For DMIS 2018 and DMIS 2020, the recommended approach is to use at least DMIS 2018 SP07 or DMIS 2022 SP03, respectively, rather than an older SP plus TCI.

1.2 Security Information

Different security considerations apply depending on the release level of the source system you are planning to use:

SAP S/4HANA

- SAP S/4HANA 2022 and higher: see SAP Note 3100673
- SAP S/4HANA 2021: implement TCI 3115128 and consider the information provided in SAP Note 3100673
- SAP S/4HANA 2020: implement TCI 3105880 and consider the information provided in SAP Note 3100673
- SAP S/4HANA 1909: implement TCI 3105890 and consider the information provided in SAP Note 3100673

DMIS 2011

- DMIS 2011 SP22 and higher: see SAP Note 3100673
- DMIS 2011 SP20 and SP21: see SAP Note 2981615
- DMIS 2011 SP19 and lower: upgrade to the latest available DMIS 2011 SP

DMIS 2018

- DMIS 2018 SP07 and higher: see SAP Note 3100673
- DMIS 2018 SP06: implement TCI 3110660 and consider the information provided in SAP Note 3100673
- DMIS 2018 SP05 and lower: upgrade to the latest available DMIS 2018 SP

DMIS 2020

- DMIS 2020 SP03 and higher: see SAP Note 3100673
- DMIS 2020 SP02 and lower: upgrade to DMIS 2020 SP03

1.3 Useful SAP Notes and Blogs

Additional information, information that becomes available on short notice, and information that is only relevant for specific release combinations is provided in SAP Notes and blog articles, in particular the following:

SAP Notes

- **Central Note**: For a compilation of the most recent information related to ABAP integration and links to the relevant SAP Notes, see SAP Note 2890171, in particular the chapter called **Important Considerations**.
- **Connection type**: For more information about connection type `ABAP` for Data Intelligence, see SAP Note 2835207.
- **Connection issues**: If the connection check for an ABAP connection fails, see SAP Note 2849542 for possible root causes and solutions.
- **Replication flows**: For known limitations and important considerations regarding replication flows, see SAP Note 3223810.

Blogs

- Beginners Guide: How to start wit ABAP Integration within Data Intelligence
- SAP Data Intelligence ABAP integration with Generation 2 - Part 1 Overview and Introduction
- SAP Data Intelligence ABAP Integration with Generation 2 - Part 2: Sample Scenario to Replicate Data from a CDS View to Google Cloud Storage
- Integrating ABAP Function modules with SAP Data Intelligence
- ABAP CDS replication in SAP Data Intelligence

- Replication and filtering of data by using SLT and SAP Data Intelligence (focus on pipelines / graphs)
- How to use SAP Data Intelligence with SAP Cloud Connector
- What's New in SAP Data Intelligence – Central Blog Post

1.4 Use Cases

You use the ABAP integration in SAP Data Intelligence Cloud features to access data from ABAP-based SAP systems, to access metadata of objects (in the SAP Data Intelligence Cloud Metadata Explorer) and to work with operators based on the ABAP programming language.

> i Note
>
> For an up-to-date list of what exactly is possible for different release combinations, see SAP Note 2890171.

Acessing the Data From ABAP-Based SAP Systems

You can use operators (generation 1 and 2) as well as the replication feature in SAP Data Intelligence Cloud to access data from ABAP-based SAP systems (SAP S/4HANA, SAP S/4HANA Cloud, SAP BW and others as specified in SAP Note 2809171).

Accessing the Metadata of Supported Objects

You can access metadata for objects from ABAP-based SAP systems in the Metadata Explorer of SAP Data Intelligence Cloud. To do so, you use the connection type ABAP, which works for CDS views, tables, and ODP objects. In addition, there is also a connection type called ABAP LEGACY; you can use it to work with ODP objects if you cannot use the connection type ABAP (for example because you cannot install the DMIS add-on).

Working With Operators Based on the ABAP Programming Language

In addition to the operators for accessing data (as described above), ABAP operators for other purposes are available, for example the generation 1 Custom ABAP Operator. For more information, see SAP Note 2809171 and Operators [page 13].

1.5 Data Extraction

The general process for data extraction with an ABAP-based SAP system as the source is the same as for any other data source in SAP Data Intelligence Cloud. However, there are a few specific aspects to be considered and **prerequisites** to be met:

- The participating systems must meet the minimum release requirements for your use case as defined in SAP Note 2890171.
- SAP Data Intelligence Cloud has its own connection type for use with ABAP-based source systems. For more information about it, see the description under ABAP and SAP Note 2835207.
- If you use an SAP S/4HANA system, you can extract data based on CDS views. (This is the recommended approach.) If you use an older system (that does not have CDS views), or if your use case requires extracting data from an SAP S/4HANA system at table level, you can alternatively extract data based on tables. In this case, you use SAP Landscape Transformation Replication Server (SLT) technology to perform the data extraction.
- To be able to access the scenario-related SAP system functionality, you need to include the required function modules in the allow list using 'UCON' for WebSocket RFC. For a list of relevant function modules, see SAP Note 2835207.
- If you want to use an SAP S/4HANA Cloud system as the source, consider the information about integration provided under Data Integration.

Related Information

Setting Up Connections [page 7]

Enabling the Data Source [page 9]

Data Type Mapping and Conversion [page 12]

Information for System Administration [page 13]

1.5.1 Setting Up Connections

For establishing the connection between your source system and SAP Data Intelligence Cloud, you can use either Remote Function Call (RFC) or WebSocket RFC.

The process for setting up a connection to SLT and to an SAP S4/HANA system is the same, as both are based on SAP NetWeaver. When setting up the connection, consider the following:

RFC

- Make sure that the connection works by sending a ping from SAP Data Intelligence Cloud to the SAP application server infrastructure on the ports of the SAP application server infrastructure.

- Create or reuse a communication or dialog user that has the necessary authorizations for accessing application tables in the SAP system as well as the necessary connectivity privileges in SAP Data Intelligence Cloud (such as SAP_DH_ABAP_MASTER, SAP_IUUC_REPL_ADMIN).
- Consider the security-related information that is provided with your version of SAP Data Intelligence Cloud.
- Use the connection management in SAP Data Intelligence Cloud to establish a connection to the source SAP system. If possible, establish a naming convention for the connection names, for example SID_MANDT_PROTOCOL (so that a connection would be called, for example, P25_001_RFC). Secure Network Communication (SNC) is available as an option.

WebSocket RFC

In addition to the prerequisites described for RFC above, the following prerequisites must be met:

- Make sure that the corresponding service infrastructure is set up correctly.
- Make a note of the configured port (which is usually the HTTPS port, see transaction SMICM in your source SAP system).
- `rfc/websocket/external_active` must be enabled to allow external RFC over WebSocket RFC.
- To ensure that only allowed function modules can be used (which is recommended), activate UCON for WebSocket RFC (ucon/websocketrfc/active) and include the relevant function modules in the UCON allowlist.

Further Considerations

- Make sure that an ALIAS has been defined for the relevant user name, and this ALIAS has been entered as a user in the Connection Management in SAP Data Intelligence.
- If you want to extract data from an ABAP-based on-premise SAP system into SAP Data Intelligence Cloud, you need to use Cloud Connector (SCC).
- If you want to use an ABAP CDS pipeline in SAP Data Intelligence Cloud to subscribe to public CDS views from an SAP S/4HANA Cloud system and consume data from these views, you need to use the integration scenario SAP_COM_0532.

Related Information

Establishing a Connection Using Cloud Connector [page 9]

1.5.1.1 Establishing a Connection Using Cloud Connector

If you want to use Cloud Connector to establish the connection between your source system and SAP Data Intelligence Cloud, there are the following additional aspects to be kept in mind:

- Consider the general information provided under SAP BTP Connectivity in the SAP Help Portal as well as the additional information about Using SAP Cloud Connector Gateway in the context of SAP Data Intelligence Cloud.
- It is important that the SSC gateway is registered with SAP Data Intelligence Cloud. (This should happen automatically when you run SAP Data Intelligence Cloud.)
- To be able to use an ABAP connection with Secure Network Communication (SNC) with Cloud Connector, you need to configure SNC in Cloud Connector (not in the connection management for SAP Data Intelligence Cloud).

To be able to work with Cloud Connector (SCC), you need to do the following:

- Install and configure SCC using SAP Cloud Connectivity Platform (as described under SAP BTP Connectivity → Integration). When doing so, consider the information provided in the installation guide and the configuration for Cloud Connector.
- Add the SCC connection parameter `Gateway` in the connection management for SAP Data Intelligence Cloud.
- In SCC, grant the functional access to the SAP NetWeaver ABAP function modules: Select the virtual host you want to use. A new pane called Resources of [your host] opens. On the new pane, choose the + symbol to add the required resources:
  - For extracting data using CDS view extraction, add DHAMB_/Prefix, DHAPE_/Prefix, and RFC_FUNCTION_SEARCH.
  - For extracting data based on tables with SAP LT Replication Server, add LTAMB_/Prefix, LTAPE_/Prefix, and RFC_FUNCTION_SEARCH.

1.5.2 Enabling the Data Source

The process for enabling your data source looks different depending on whether you are using CDS views, tables, or ODP as the source.

The reason for this is that the system landscape components are different: For extracting **CDS view data**, SAP Data Intelligence Cloud can directly connect to the source system and retrieve data from there. For extracting data directly from **tables**, you additionally need SAP Landscape Transformation Replication Server (SLT), which acts as an intermediary between your data source and SAP Data Intelligence Cloud. To extract data using **ODP**, SAP Data Intelligence Cloud connects directly to the source SAP system, which requires the DMIS add-on as well as the ODP Application API.

> **i Note**
>
> Consider also the important security settings described in SAP Note 3100673. (This SAP Notes describes the settings for the current releases and provides links to SAP Notes for older releases.)

CDS Views:

What you need to do to integrate data using ABAP CDS views depends on whether you want to use SAP S/4HANA or SAP S/4HANA Cloud as the data source.

For *SAP S/4HANA Cloud*, only standard ABAP CDS views that are C1-released can be integrated using SAP Data Intelligence Cloud. For more information, see CDS Views.

For SAP S/4HANA, you have the following options:

- You can integrate data from standard ABAP CDS views that have the required annotations as part of the C1 release contract (in a similar way as with SAP S/4HANA Cloud).
- In addition, you can create custom ABAP CDS views to integrate data with SAP Data Intelligence Cloud using the ABAP Development Tool (ADT Tool) as described in Loading and Replicating Data from ABAP CDS Views in SAP S/4HANA [page 11].

For more information, see also CDS Views.

To find out which extraction-enabled CDS views are C1-released and delta-enabled, you can use the public CDS View `I_DataExtractionEnabledView`, which is available in SAP S/4HANA Cloud as well as in SAP S/4 HANA (as of SAP S/4HANA 2020).

Alternatively, you can check for the following properties in the Metadata Explorer of SAP Data Intelligence Cloud:

- Property name “Extraction Enabled” for general extraction of a specific ABAP CDS view
- Property name “Delta enabled” for delta extraction of a specific ABAP CDS view
- Property name “CDS Type” for checking the contract status, e.g. “C1” for C1 released ABAP CDS views

SLT Table Replication

Proceed as follows:

1. Set up a new SLT configuration in transaction `ltrc` of the SLT system. Choose the respective source system and use SAP Data Intelligence as the target. Note that you need to use different SLT configurations depending on whether you are working with SAP Data Intelligence Cloud operators of generation 1 or generation 2, respectively, or with replication flows. For more information, see the application help for SLT.
2. Once the SLT configuration is available, make a note of its 3-digit identifier (mass transfer ID).
3. In the operator data, you can now enter the mass transfer ID as well as the name of the table you want to extract data from.

ODP

If you want to integrate systems using ODP, make sure that the following prerequisites are met:

- The respenctive DMIS version is available in your system. For more information, see SAP Note 2890171.
- The prerequisites for ODP integration with SAP Data Intelligence Cloud (as described in SAP Note 2775549) are fulfilled.

- The technical user for using the ODP interface has the required privileges assigned. For more information, see SAP Note 3100673.

Related Information

Loading and Replicating Data from ABAP CDS Views in SAP S/4HANA [page 11]

1.5.2.1 Loading and Replicating Data from ABAP CDS Views in SAP S/4HANA

To make the data from an ABAP CDS view available in SAP Data Intelligence Cloud, you need to specify specific annotations for the view in the ABAP Development Tool (ADT Tool).

> i Note
>
> This applies to SAP S/4HANA, but **not** to SAP S/4HANA Cloud.

The annotations allow the view to use a trigger-based Change Data Capture (CDC) recording mechanism to load data and record changes to the tables that belong to the view.

The relevant annotations are described in the sections below.

> i Note
>
> You can find detailed information about all the annotations mentioned in this document on the SAP Help Portal at https://help.sap.com. Search for `CDS Annotations` and select the entry for product SAP S/4HANA. Expand the node *CDS Annotations* and select the node *Analytics Annotations*.

Loading Data from a CDS View

To load data from a custom CDS view, add the following annotation to the view:

- `Analytics.dataExtraction.enabled`

> i Note
>
> Not all CDS views delivered by SAP have this annotation. You can view a list of the CDS views that already use this annotation by creating a query that uses a SELECT statement for the following CDS view:
>
> - `I_DataExtractionEnabledView`

Alternatively, you can use the data preview function in the SAP HANA Studio or ABAP Development tools for Eclipse.

Replicating Data from a CDS View

The Change Data Capture recording mechanism uses database triggers to record any changes to the tables that belong to an ABAP CDS view. To do this, the key fields of all underlying tables need to be mapped to the fields of the CDS view. For simple CDS views such as CDS projection views, this can be done automatically by using the following annotation:

- `Analytics.dataExtraction.delta.changeDataCapture.automatic`

For more complex CDS views that use joins, the mapping must be done manually. You use the following annotation to do this:

- `Analytics.dataExtraction.delta.changeDataCapture.mapping`

Definition of Annotations

The following code shows the options that are available for the annotations described above.

Sample Code

```
Annotation Analytics {
  dataCategory : String(20) enum { DIMENSION; FACT; CUBE; AGGREGATIONLEVEL; };
  dataExtraction: {
    enabled : Boolean default true;
    delta : {
      changeDataCapture : {
        automatic : Boolean default true;
        mapping : {
          role : String(30) enum {MAIN; LEFT_OUTER_JOIN_TO_ONE_JOIN;};
          table : String(30);
          viewElement : array of ElementRef;
          tableElement : array of ElementRef;
          filter : {
            tableElement : ElementRef;
            operator : String(11) enum
{EQ;NOT_EQ;GT;GE;LT;LE;BETWEEN;NOT_BETWEEN;} default #EQ;
            value : String(45);
            highValue : String(45);
                    };
              };
            };
          };
         };
```

1.5.3 Data Type Mapping and Conversion

When extracting data from an ABAP-based source system and transferring it to a target system, it is not always possible to keep the data content and source format as-is (for technical reasons). Where necessary, the ABAP data types from the source are automatically replaced with their respective string representations during the data transfer (This is also known as wire format conversion.)

For some of the **generation 1** ABAP operators, you can choose between the following options:

- **Enhanced format conversion**: This conversion type not only changes the technical type (which is necessary to allow a transfer at all), but also validates the data. For example, values that the system considers "invalid" with respect to the applicable ISO standards are sent to the target as "NaN" (Not a number) or "undefined".
- **Required conversions**: This conversion type changes some technical types without validation. For example, values that the system considers as invalid with respect to the applicable ISO standards are sent to the target without conversion, but as valid values. This version includes changes that are necessary to enable the transfer and cannot be skipped (minimum scope).

- **Required conversions plus currency**: This conversion type changes some technical types without validation (as described above for required conversions) and converts currency values with regard to currency shift.
- **Required conversions plus time format and currency**: This version of the wire format conversion changes some technical types without validation (as described above for required conversions), converts currency values respecting currency shift, and renders date or time formats in ISO format.

What options exactly you have depends on the available operator versions, and these in turn depend on the SAP S/4HANA version (or DMIS version, respectively) of your ABAP-based SAP system (not on your version of SAP Data Intelligence.) For more information, see the descriptions of the individual operators in the Repository Object Reference for SAP Data Intelligence.

For the **generation 2** operator Read Data From SAP System and for replication flows, the conversion type Required Conversions Plus Time Format and Currency is used by default and cannot be changed.

For more information, including lists of the source data types that are changed during the transfer as well as the corresponding target formats, see SAP Note 3035658.

1.5.4 Information for System Administration

Some aspects of system administration for SAP Data Intelligence are specific to ABAP integration. In particular, this is true for ABAP connection provider (also known as Axino):

Axino is a feature for establishing connections between remote ABAP-based SAP systems and SAP Data Intelligence and enabling bidirectional data transfer. As a rule, Axino runs in the background using default settings, and there is no user interaction required. If necessary, however, you can change some settings in the System Management app of SAP Data Intelligence as follows:

- Using **core dump files** in connection with Axino issues: If Axino crashes and you are unable to reproduce the issue or find a root cause, you can enable generation of core dump files and send the resulting files to SAP for further analysis. For more information as well as a detailed how-to description, see SAP Note 3033853.
- **Auto-scaling**: As a rule, Axino runs in the background in one service instance of SAP Data Intelligence and uses default settings, and there is no user interaction required. However if the number of workload requests becomes very high, the available CPU or memory capacity may no longer be sufficient, and as a consequence subsequent requests run into performance issues and 'out of memory' errors. If this happens, you can try switching on the scale-out feature to use multiple service instances and balance the workload. For further information, see SAP Note 3148794.
- If you want to use **Secure Network Communication** (SNC), consider the information provided under Configure Secure Network Communication for ABAP.

1.6 Operators

The ABAP operators differ from other types of operators in that their implementations exist in their own repository (called ABAP pipeline engine) in the ABAP-based SAP system.

The modeler in SAP Data Intelligence Cloud contains a shell for each impementation of these operators so that you can work with them and include them in graphs (pipelines) in the same way as any other operators. In order to retrieve the relevant implementations from the ABAP-based SAP system, you need to provide an RFC connection. The ABAP pipeline engine is included in the standard shipments for SAP S/4HANA Cloud and SAP S/4HANA (as of SAP S/4HANA 1909). For older releases (NetWeaver 7.0 or higher), you need to install the DMIS Add-on to get the ABAP pipeline engine.

Operator Generations and Versions

Two generations of operators are available. The generation 2 operators differ from the generation 1 operators in substantial ways. (For example, the subscription feature has been replaced with a resilience feature.) As a consequence, it is not possible to combine generation 1 and generation 2 operators in one graph.

The existing versioning concept remains in place and is independent of operator generations.

For a descripton of all available operators, see the Repository Objects Reference for SAP Data Intelligence Cloud:

- Generation 1 (ABAP)
- Generation 2 (ABAP)

Related Information

Creating a Custom ABAP Operator [page 14]

1.6.1 Creating a Custom ABAP Operator

This text describes the steps you need to perform in your ABAP-based SAP system to implement your own custom ABAP operator.

Context

> ℹ Note
>
> - Custom operators can only be used together with the Custom ABAP Operator, which is a generation 1 operator in SAP Data Intelligence.
> - This text describes how to create a custom ABAP operator in the ABAP-based SAP system. For general information about how to work with custom operators, see Creating Operators in the Modeling Guide for SAP Data Intelligence.

For more information, see also the following blog article: https://blogs.sap.com/2021/06/01/integrating-abap-function-modules-with-sap-data-intelligence/

You create custom ABAP operators in the ABAP-based SAP system by implementing a BAdI called BADI_DHAPE_ENGINE_OPERATOR. For each BAdI implementation, a filter must be defined that matches the name of the operator (example: com.sap.abap.pass).

The BAdI implementation consists of a class with two methods that must be redefined. The BAdI implementation should extend the abstract class cl_dhape_graph_oper_abstract.

- GET_INFO: Returns metadata about the operator
- NEW_PROCESS: Creates a new instance of the operator

The operator instance (process) implements the interface if_dhape_graph_process, though it is recommended that the process extends the abstract class cl_dhape_graph_proc_abstract instead.

Creating the artefacts for your ABAP operator: To reduce manual activities to a minimum, there is a framework that supports you in creating all the described artefacts in the ABAP backend. The framework consists of two reports that must be executed in sequence. They are available with SAP S/4HANA 1909 and any SAP NetWeaver-based system with SAP_BASIS 7.76 and higher or SAP_ABA 75E and higher.

Procedure

1. Execute the DHAPE_CREATE_OPERATOR_CLASS report.

   The report creates the operator implementation class of your custom ABAP operator.

   > **i Note**
   >
   > If your ABAP-based SAP system is not an S/4HANA System, use report R_LTAPE_CREATE_OPERATOR_CLASS to create the operator implementation class.

   You must provide the following parameters:

   - Class name
   - Operator name in SAP Data Intelligence
   - Class description
2. Execute the DHAPE_CREATE_OPER_BADI_IMPL report.

   > **i Note**
   >
   > If your ABAP-based SAP system is not an S/4 HANA system, use report R_LTAPE_CREATE_BADI_IMPL to create the operator implementation class.

   You must provide the following parameters:

   - Enhancement impl. name
   - Class description
   - BAdI Implementation
   - Reference to impl. class
   - Operator name (SAP Data Intelligence)

   Now the class is created and the two methods GET_INFO and STEP can be modified according to your needs.
3. Adapt the GET_INFO method.

Adapt the GET_INFO method template sections of the new operator to meet your requirements:

- Change the number, names, and typing of the ports according to your needs (the structure *ls_port* in the below template code).
- Add any parameters, if needed (the structure ls_prop in the below template code).

4. Adapt the *process* local class.

   Inside the process method, you can implement the actual logic that you would like to execute within your custom ABAP operator. Processes use a simple event-based model and can be implemented by redefining one or more of the following methods:

   - ON_START: Called once before the graph is started.
   - ON_RESUME: Called at least once before the graph is started or resumed.
   - STEP: Called frequently.
   - ON_SUSPEND: Called at least once after the graph is stopped or suspended.
   - ON_STOP: Called once after the graph is stopped.

   Additionally, the abstract class provides several helper methods that you can use:

   - GET_PORT: Get an input or output port.
   - GET_CONF_VALUE: Get a value from the process configuration properties.

Example

To become more familiar with the event-based model, use this example implementation:

Sample Code

```
CLASS lcl_process DEFINITION
INHERITING FROM cl_dhape_graph_proc_abstract.
PUBLIC SECTION.
METHODS: if_dhape_graph_process~on_resume REDEFINITION.
METHODS: if_dhape_graph_process~step REDEFINITION.
PRIVATE SECTION.
DATA: mo_in TYPE REF TO if_dhape_graph_channel_reader.
DATA: mo_out TYPE REF TO if_dhape_graph_channel_writer.
DATA: mv_data TYPE string.
ENDCLASS.
CLASS lcl_process IMPLEMENTATION.
METHOD if_dhape_graph_process~on_resume.
mo_in = get_port( 'in' )->get_reader( ).
mo_out = get_port( 'out' )->get_writer( ).
ENDMETHOD.
METHOD if_dhape_graph_process~step.
CHECK mv_alive = abap_true.
IF mo_in->has_data( ).
CHECK mo_out->is_blocked( ) <> abap_true.
mo_in->read_copy( IMPORTING ea_data = mv_data ).
mo_out->write_copy( reverse( mv_data ) ).
ELSEIF mo_in->is_closed( ).
mo_out->close( ).
mv_alive = abap_false.
ENDIF.
rv_progress = abap_true.
ENDMETHOD.
ENDCLASS.
```

In the ON_RESUME method, you get the input and output channels of the operator. This only needs to be done once.

> **i Note**
>
> If the user tries to get a port that is not defined in GET_INFO or tries to get a reader for an output port that does not exist, this results in an error.

In the STEP method, the operator first checks if the process is still alive. If this is not the case, no further action is taken.

The operator then checks if there is any data available on the input channel. If this is the case, and if the output channel is not blocked, the operator reads the data from the input channel. Finally, it reverses the data and writes it to the output channel.

Until the data is consumed by a subsequent process, the output is blocked.

If the input is empty and already closed, the system closes the output channel and sets the operator to be no longer alive. Once all operator processes indicate that they are no longer alive, the graph terminates automatically.

The system may temporarily suspend graph execution, for example during a maintenance event. When this happens, you can store any necessary local state by adding it to the io_saved_state object that is passed to the ON_SUSPEND method.

When the graph is resumed later, the same io_saved_state object is again passed to the ON_RESUME method. You can restore the local state as described above.

To get the new operator definition into the repository of SAP Data Intelligence, use custom ABAP operator.

1.7 Troubleshooting

Here you can find a list of known issues with ABAP integration in SAP Data Intelligence and their solutions.

- You have started a pipeline with ABAP operators in SAP Data Intelligence and get an error message "Number of sessions exceeded" (or similar).
  Solution: This issue is related to the maximum number of sessions in ABAP Pipeline Engine. For more information, see SAP Note 2999448.
- ABAP connection type check (in the connection management for SAP Data Intelligence fails.
  Solution: see SAP Note 2849542.
- You are trying to extract a CDS view in a generation 1 pipeline using the CDS Reader operator and receive an error message in the process logs like “CDS view <CDS View name> does not support data extraction”.
  Solution: Make sure that you extract a CDS view that is enabled for extraction as well as delta extraction, for example by using the required annotations (for a custom CDS view) or choosing a standard CDS view that is enabled for extraction. See also Enabling the Data Source [page 9].
- A generation 1 pipeline that includes the SLT Connector operator fails with an error message like: “Subscription is already being used by another graph”.
  Solution: see SAP Note 3057246
- A generation 1 graph fails with an error message like “The graph failed with error "invalid character '\xXX' in string literal""

Solution: see SAP Note 3016338

- Table extraction using the generation 2 operator Read Data From SAP System fails, and you get an error message like: “Object does not exist in the source system”.
  Solution: see SAP Note 3143151

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.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap

© 2022 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.

THE BEST RUN SAP