Informatica®

Informatica® Cloud Data Integration

SAP Table Connector

Informatica Cloud Data Integration SAP Table Connector
July 2024

© Copyright Informatica LLC 2023, 2024

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Informatica, the Informatica logo, Informatica Cloud, and PowerCenter are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at https://www.informatica.com/trademarks.html. Other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties. Required third party notices are included with the product.

See patents at https://www.informatica.com/legal/patents.html.

DISCLAIMER: Informatica LLC provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability, or use for a particular purpose. Informatica LLC does not warrant that this software or documentation is error free. The information provided in this software or documentation may include technical inaccuracies or typographical errors. The information in this software and documentation is subject to change at any time without notice.

NOTICES

This Informatica product (the "Software") includes certain drivers (the "DataDirect Drivers") from DataDirect Technologies, an operating company of Progress Software Corporation ("DataDirect") which are subject to the following terms and conditions:

1. THE DATADIRECT DRIVERS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.
2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at infa_documentation@informatica.com.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2024-07-14

Table of Contents

Preface 6

Informatica Resources 6

- Informatica Documentation 6
- Informatica Intelligent Cloud Services web site 6
- Informatica Intelligent Cloud Services Communities 6
- Informatica Intelligent Cloud Services Marketplace 6
- Data Integration connector documentation 7
- Informatica Knowledge Base 7
- Informatica Intelligent Cloud Services Trust Center 7
- Informatica Global Customer Support 7

Chapter 1: Introduction to SAP Table Connector 8

SAP objects 8

- ABAP CDS views 9
- Rules and guidelines for SAP sources and targets 9

SAP ADSO source objects 11

- Naming conventions of SAP ADSO objects 11

SAP Table Connector assets 12

Chapter 2: Connections for SAP Table Connector 13

Prerequisites 13

- Download and configure the SAP libraries 13
- Configure SAP user authorization 15
- Install transport files to read from an SAP table 16
- Install transport files to write to an SAP table 17

Connect to SAP table 17

- Before you begin 17
- Connection details 18

Configure the sapnwrfc.ini file 20

Configure HTTPS to connect to SAP 22

- Create an OpenSSL certificate 23
- Convert an OpenSSL certificate to PSE format 24
- Enable the HTTPS service on the SAP system 25
- Import the certificate to the SAP system trust store 25

Configure the Secure Network Communication protocol 25

Enable the Secure Agent to operate as a whitelisted host in SAP (Optional) 26

Use the serverless runtime environment 27

Troubleshooting an SAP Table connection 28

Chapter 3: Mappings and mapping tasks with SAP Table. 29

SAP Table sources in mappings. 29
Filter options. 31
Sort options. 35
Join conditions. 36
Advanced properties. 36
Delta extraction for SAP table reader mappings. 37
Delta extraction behavior. 37
Informatica custom table /INFADI/TBLCHNGN. 38
Update modes for delta extraction 38
Configuring a parameter file for delta extraction. 40
Configure a table name override for delta extraction 41
Rules and guidelines for delta extraction. 41
Configuring delta extraction for an SAP table reader mapping. 42
Troubleshooting delta extraction for SAP Table Reader mappings. 42
Key range partitioning for SAP Table sources. 43
Configuring key range partitioning for SAP Table sources. 44
Best practices for key range partitioning. 44
SAP Table lookups in mappings. 44
Configuring a mapping with an SAP Table source. 44
Creating a mapping task. 46
Mapping with an SAP Table source example. 47
Step 1: Define the mapping. 47
Step 2: Configure the SAP Table source. 47
Step 3: Configure the flat file target. 50
Step 4: Save the mapping and create a mapping task. 52

Chapter 4: Synchronization tasks with SAP Table. 53

SAP Table sources in synchronization tasks. 53
SAP Table lookups in synchronization tasks. 54
Configuring a synchronization task with a single SAP object as the source. 55
Configuring a synchronization task with multiple SAP objects as the source. 57
Monitoring a synchronization task. 58
Synchronization task example. 59
Step 1: Define the synchronization task. 59
Step 2: Configure the SAP Table source. 60
Step 3: Configure the flat file target. 62
Step 4: Configure the field mapping. 63

Chapter 5: Data type reference. 64

SAP and transformation data types. 64
Rules and guidelines for SSTRING, STRING, and RAWSTRING data types. 66

Chapter 6: FAQ for SAP Table Connector. 67

Index. 68

Preface

Use SAP Table Connector to learn how to read from or write to SAP by using Cloud Data Integration. Learn to create an SAP Table Connector connection, develop and run synchronization tasks, mappings, mapping tasks, and data transfer tasks in Cloud Data Integration.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit https://docs.informatica.com.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica Intelligent Cloud Services web site

You can access the Informatica Intelligent Cloud Services web site at http://www.informatica.com/cloud. This site contains information about Informatica Cloud integration services.

Informatica Intelligent Cloud Services Communities

Use the Informatica Intelligent Cloud Services Community to discuss and resolve technical issues. You can also find technical tips, documentation updates, and answers to frequently asked questions.

Access the Informatica Intelligent Cloud Services Community at:

https://network.informatica.com/community/informatica-network/products/cloud-integration

Developers can learn more and share tips at the Cloud Developer community:

https://network.informatica.com/community/informatica-network/products/cloud-integration/cloud-developers

Informatica Intelligent Cloud Services Marketplace

Visit the Informatica Marketplace to try and buy Data Integration Connectors, templates, and mapplets:

https://marketplace.informatica.com/

Data Integration connector documentation

You can access documentation for Data Integration Connectors at the Documentation Portal. To explore the Documentation Portal, visit https://docs.informatica.com.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit https://search.informatica.com. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Intelligent Cloud Services Trust Center

The Informatica Intelligent Cloud Services Trust Center provides information about Informatica security policies and real-time system availability.

You can access the trust center at https://www.informatica.com/trust-center.html.

Subscribe to the Informatica Intelligent Cloud Services Trust Center to receive upgrade, maintenance, and incident notifications. The Informatica Intelligent Cloud Services Status page displays the production status of all the Informatica cloud products. All maintenance updates are posted to this page, and during an outage, it will have the most current information. To ensure you are notified of updates and outages, you can subscribe to receive updates for a single component or all Informatica Intelligent Cloud Services components. Subscribing to all components is the best way to be certain you never miss an update.

To subscribe, on the Informatica Intelligent Cloud Services Status page, click **SUBSCRIBE TO UPDATES**. You can choose to receive notifications sent as emails, SMS text messages, webhooks, RSS feeds, or any combination of the four.

Informatica Global Customer Support

You can contact a Global Support Center through the Informatica Network or by telephone.

To find online support resources on the Informatica Network, click **Contact Support** in the Informatica Intelligent Cloud Services Help menu to go to the **Cloud Support** page. The **Cloud Support** page includes system status information and community discussions. Log in to Informatica Network and click **Need Help** to find additional resources and to contact Informatica Global Customer Support through email.

The telephone numbers for Informatica Global Customer Support are available from the Informatica web site at https://www.informatica.com/services-and-training/support-services/contact-us.html.

CHAPTER 1

Introduction to SAP Table Connector

You can integrate data with SAP at the data-level by using SAP Table Connector.

To read data from SAP tables and write data only to SAP custom tables that have been created under the customer namespace, you can use an SAP Table Connector connection.

You can read data from transparent tables, cluster tables, pool tables, views, and ABAP CDS views. The Secure Agent accesses data through the application layer in SAP using ABAP. Data is streamed to the Secure Agent through the HTTP or HTTPS protocol. SAP Table Connector supports joins and filters on the source tables.

You can also use SAP Table Connector to read data from an SAP ADSO object. To write data to an ADSO, use SAP ADSO Writer Connector.

To optimize performance when the Secure Agent and the SAP system are in different networks, you can enable data compression when you read data from SAP.

When you create a synchronization task, mapping, mapping task, or data transfer task, Data Integration generates a dynamic ABAP query to read from SAP tables and write to custom SAP tables. You can switch mappings to advanced mode to include transformations and functions that enable advanced functionality.

Note: When you want to write data to SAP standard tables, SAP recommends that you use OData, BAPI/RFC, or IDoc API to update data within the SAP application. For information about using an SAP Table Connector connection to write data to SAP custom tables, contact Informatica Global Customer Support.

SAP objects

You can connect to SAP views, ABAP CDS views, transparent, pool and cluster tables using an SAP Table connection.

Data Integration does not differentiate between tables and views. You extract data from views the same way you extract data from tables. When you select a table, Data Integration displays the table name followed by the business name in the **Select Object** dialog box. You can filter by table name or business name when you connect to the SAP system.

Data Integration imports the following SAP table information:

- Source name
- Column names
- Business descriptions

- Data types, length, precision, and scale

ABAP CDS views

You can use an ABAP CDS view as a source or lookup in mappings.

Note: ABAP CDS views are supported from SAP NetWeaver system version 7.50 SP4 or later.

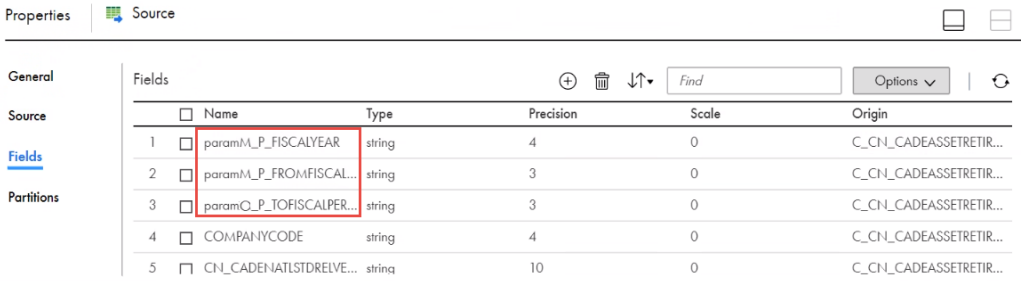
When you import an ABAP CDS view into Cloud Data Integration, the agent adds a prefix to the parameter name. The prefix is used to indicate the parameter type.

You can use the following types of parameters:

- **Mandatory Parameter.** A parameter for which you need to specify a value. For example, in the field `paramM_P_FISCALYEAR`, `paramM` is the prefix for a mandatory parameter that the agent adds. `P_FISCALYEAR` is the parameter name that is a part of ABAP CDS views.
- **Optional Parameter.** When you define a parameter in SAP and use the annotation `@Environment.systemField`, the parameter appears as an optional parameter in the list of fields. If you do not provide a value, the optional parameter uses the ABAP system fields values.

  For example, in the field `paramO_P`, `paramO` is the prefix for an optional parameter that the agent adds. `P_TOFISCALPER` is the parameter name that is a part of ABAP CDS views.

  The following image shows the mandatory and optional parameter when you select a CDS view:



  In the example, paramO denotes an optional parameter and paramM denotes a mandatory parameter.

Rules and guidelines for SAP sources and targets

Use the following rules and guidelines when you configure SAP sources and targets:

- When you configure an SAP source, configure row limits using the advanced source properties available on the scheduling page of the task wizard. Row limits on the data filters page of the task wizard are not enabled for SAP sources.
- Do not use tables as SAP Table sources if the sources have circular primary key-foreign key relationships.
- When you use more than one SAP table in a synchronization task, you can use one cluster table or one pool table. If you use more than one cluster or pool table, errors occur at run time. You can use the Object Search dialog box to see if a table is a cluster table or a pool table. You can use multiple transparent tables in a task.
- When you join a cluster table or pool table with a transparent table, include all key fields in the transparent table in the join condition. List the fields in the order that they appear in the SAP system.
- When you join a cluster table or pool table with a transparent table, use all of the source fields in the transparent table that you use in the joins and filters in the field mapping. Also, map at least one field from the cluster or pool table.

- Define relationships for multiple sources after the data preview displays the data. You can use the wizard in advanced mode to avoid waiting to preview data.
- Data sorting does not apply on cluster or pool table fields.
- You can use Data Integration variables and ABAP variables in simple data filters for SAP tables. Do not use ABAP syntax in simple data filters for SAP tables.
- You can use ABAP variables and ABAP syntax in advanced data filters for SAP tables. Do not use Data Integration variables in advanced data filters for SAP tables.
- Use the $LastRunTime variable to filter data based on the start date and time in GMT time zone of the last task that ran successfully or ended with a warning in the Secure Agent.
- Due to an SAP limitation, tasks that require a read longer than 30 minutes can fail. You might use one or more of the following suggestions if you encounter this problem:
  - Use the SAP advanced source properties to limit the number of rows to be read.
  - Configure a data filter to reduce the number of rows to be read.
  - Reduce the number of output fields for the task.
  - Configure the SAP parameter rdisp/max_wprun_time to allow more time for the read. For more information, see the SAP documentation.
  - To increase the amount of records that the Secure Agent can retrieve at one time, you can increase the Java heap memory for the Secure Agent. To do this, edit the Secure Agent. In the System Configuration Details section, select DTM and set the JVMOption1 property to the following value: Xmx512m. Click OK to save the change and restart the Secure Agent. Adjust the value for the JVMOption1 property based on the amount of records you want to retrieve and the available memory on the Secure Agent machine.
- For a lookup on an SAP object, configure the lookup to return less than 20 rows. Tasks might fail if the lookup returns more than 20 rows.
- A lookup on an SAP object does not return matching rows if the lookup comparison value is NULL.
- When you define a reject file name for an SAP target, use the default name or the variable $ErrorFileName. The $ErrorFileName variable uses the following convention for reject file name:

  ```
  s_dss_<task name>_<run number>_error.csv.bad
  ```
- When you define a reject directory for an SAP target, use the variable $PMBadFileDir. When you use the $PMBadFileDir variable, the synchronization task writes the reject file to the following Secure Agent directory:

  ```
  <SecureAgent_InstallDir>/main/rdtmDir/error
  ```
- Consider the following rules when you configure a mapping to read from CDS views:
  - When you select a CDS view as an SAP source object in a mapping, you cannot preview the data.
  - When you configure a mapping with a lookup transformation to look up records in CDS views with parameters, only uncached lookup is supported.
  - Delta extraction does not apply if you select a CDS view as an SAP source object.
  - Do not use completely parameterized or advanced filters for CDS view parameters.
  - Do not use the CLNT field optional parameter in a mapping.
- To establish communication from the SAP system with the Secure Agent using the IP address of the NAT gateway, you need to add the DTM property named SapTableReaderNatIpAddress for the Secure Agent and specify the NAT IP address as the value.
- You cannot configure a custom relationship that uses a right outer join for SAP Table objects from the SAP NetWeaver system version 7.40 SP04 or earlier.
- When you run an SAP Table Reader mapping, even though the mapping runs successfully, the system log of the SAP system displays the CPIC error.

To avoid the CPIC error at the SAP system, see KB article 000176711.

- You can't configure multiple source objects in a parametrized mapping task, or a mapping task that uses the following mapping task templates:
  - Replicate an SAP table incrementally into a new target.
  - Replicate an SAP table incrementally into an existing target.

SAP ADSO source objects

You can use SAP Table Reader Connector to connect to the SAP BW/4HANA application and read large volumes of data from SAP ADSO objects.

SAP Advanced DataStore Object (ADSO) is the central object for data storage and data consolidation in the SAP BW system. ADSO is the primary persistent object in BW/4HANA and simplifies the modeling and support environment. ADSOs are commonly stored in 2-dimensional transparent database tables.

Data Integration imports the following SAP ADSO information:

- ADSO name
- Description
- Type
- Info area
- Transfer structure
- Field information such as the name, type, description, precision, scale, SAP type, SAP length, SAP output length, and SAP decimals.

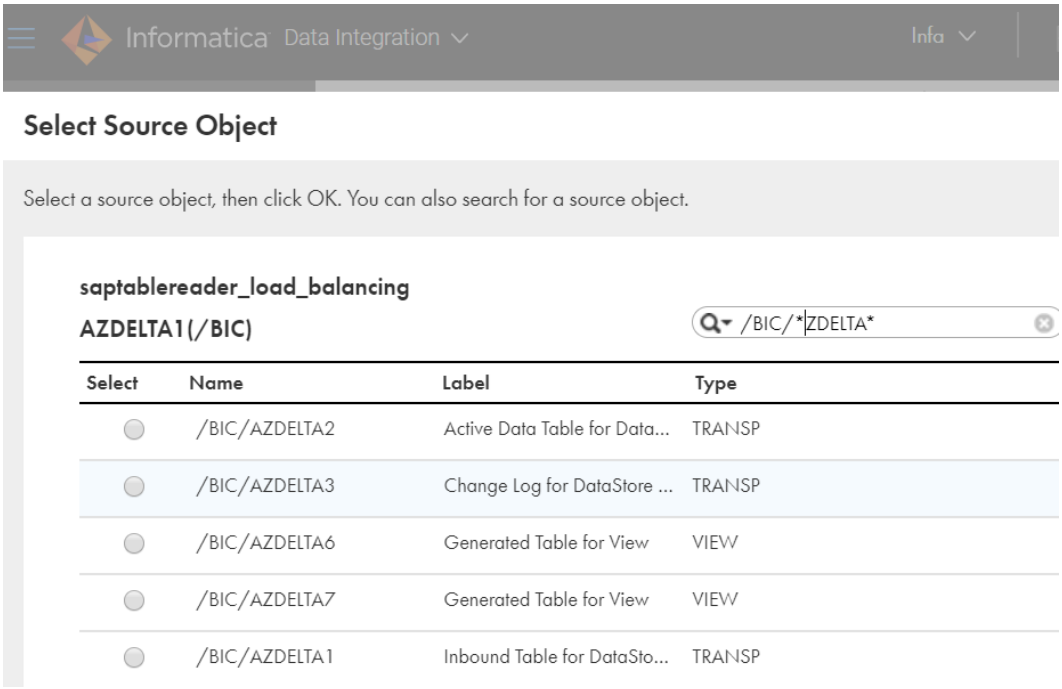
Naming conventions of SAP ADSO objects

When you import an ADSO object as a source in Data Integration mappings, it follows `/BIC/A<technical_name of ADSO>*` naming convention, where * represents the table number associated with an ADSO.

When you import an ADSO into Cloud Data Integration, the Secure Agent imports the following five tables associated with each ADSO:

- `/BIC/A<technical_name of ADSO>1`. Inbound table for data store <technical_name of ADSO>.
- `/BIC/A<technical_name of ADSO>2`. Active data table for data store <technical_name of ADSO>.
- `/BIC/A<technical_name of ADSO>3`. Change log for data store <technical_name of ADSO>.
- `/BIC/A<technical_name of ADSO>6`. View for extraction from data store <technical_name of ADSO>.
- `/BIC/A<technical_name of ADSO>7`. View for reporting for datastore <technical_name of ADSO>.
- `/BIC/A<technical_name of ADSO>8`. View for external access (introduced in BW/4HANA 2.0 <technical_name of ADSO>.

The following image shows the five tables of an ADSO with technical name ZDELTA that you can import from BW/4HANA 1.0 into Cloud Data Integration:



SAP Table Connector assets

Create assets in Data Integration to integrate data using SAP Table Connector.

When you use SAP Table Connector, you can include the following Data Integration assets:

- Data transfer task
- Mapping
- Mapping task
- Synchronization task

For more information about configuring assets and transformations, see *Mappings*, *Transformations*, and *Tasks* in the Data Integration documentation.

CHAPTER 2

Connections for SAP Table Connector

Create an SAP Table Connector connection to access data from SAP tables and SAP ADSO objects. You can use an SAP Table connection in synchronization tasks, mappings, and mapping tasks.

You can use an SAP Table connection to read data from the following objects:

- Transparent tables
- Cluster tables
- Pool tables
- Views
- Views
- SAP ADSO

You can also use the SAP Table connection to write data to custom transparent tables.

Prerequisites

Before you use an SAP Table connection, the SAP administrator needs to perform certain prerequisite tasks to configure the Secure Agent machine and SAP system.

To process data through SAP, you also need to verify if the required licenses are enabled for the SAP system.

Download and configure the SAP libraries

To read data from or write data to SAP tables, you need to download and configure the SAP NetWeaver RFC SDK libraries and SAP JCo libraries on the Secure Agent machine. If you encounter any issues while you download libraries, contact SAP Customer Support.

1. Go to the SAP Support Portal, and then click **Software Downloads**.

   Note: You need to have SAP credentials to access **Software Downloads** from the SAP Support Portal.
2. Download the latest version of the SAP NetWeaver RFC SDK 7.50 libraries that are specific to the operating system that hosts the Secure Agent.

The following table lists the libraries corresponding to the different operating systems:

Operating System	SAP NetWeaver RFC SDK Libraries
Linux 64	- libicudata.so.50 - libicui18n.so.50 - libicuuc.so.50 - libsapnwrfc.so - libsapucum.so
Windows 64	- icudt50.dll - icuin50.dll - icuuc50.dll - libsapucum.dll - sapnwrfc.dll

3. Copy the SAP NetWeaver RFC SDK 7.50 libraries to the following directory:
   `<Informatica Secure Agent installation directory>\apps\Data_Integration_Server\ext\deploy_to_main\bin\rdtm`

   Create the `deploy_to_main\bin\rdtm` directory if it does not already exist.
4. Set the following permissions for each NetWeaver RFC SDK library:
   - Read, write, and execute permissions for the current user.
   - Read and execute permissions for all other users.
5. From the SAP Support Portal, download the latest version of the 64-bit SAP JCo libraries based on the operating system on which the Secure Agent runs:

Secure Agent System	SAP JCo Libraries
Windows	sapjco3.jar sapjco3.dll
Linux	sapjco3.jar libsapjco3.so

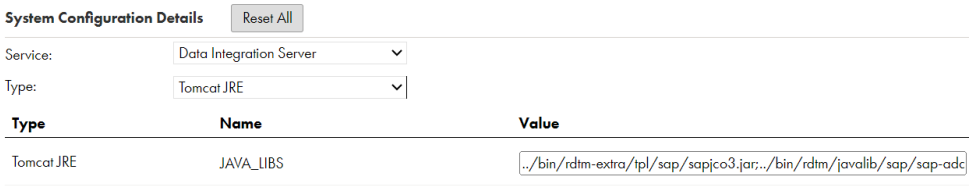
6. Copy the JCo libraries to the following directory:
   `<Informatica Secure Agent installation directory>\apps\Data_Integration_Server\ext\deploy_to_main\bin\rdtm-extra\tpl\sap`

   Create the `deploy_to_main\bin\rdtm-extra\tpl\sap` directory if it does not already exist.
7. Log in to Informatica Intelligent Cloud Services and configure the JAVA_LIBS property for the Secure Agent.
   a. Select **Administrator** > **Runtime Environments**.
   b. Click **Runtime Environments** to access the **Runtime Environments** page.
   c. To the left of the agent name, click **Edit Secure Agent**.
   d. From the **Service** list, select **Data Integration Server**.
   e. From the **Type** list, select **Tomcat JRE**.

f. Enter the JAVA_LIBS value based on the operating system on which the Secure Agent runs.

Operating System	Value
Windows	../bin/rdtm-extra/tpl/sap/sapjco3.jar;../bin/rdtm/javalib/sap/sap-adapter-common.jar
Linux	../bin/rdtm-extra/tpl/sap/sapjco3.jar:../bin/rdtm/javalib/sap/sap-adapter-common.jar

Warning: If you copy the value directly from the table, the hyphens (-) in the value might be incorrectly copied. Copy the value to a text editor and make sure that the value you copied is not corrupted.



g. Click **Save**.

h. Repeat steps 2 through 7 on every machine where you installed the Secure Agent.

8. Restart the Secure Agent.

Configure SAP user authorization

Configure the SAP user account in the SAP system to process SAP table data.

For more information about how to configure SAP user authorization in the SAP system, see SAP user authorizations.

The following table describes the required authorization to read from SAP tables:

Read Object Name	Authorization
S_BTCH_JOB	DELE, LIST, PLAN, SHOW. Set Job Operation to RELE.
S_PROGRAM	BTCSUBMIT, SUBMIT
S_RFC	SYST, SDTX, SDIFRUNTIME, /INFADI/TBLRDR, RFC1
S_TABU_DIS / S_TABU_NUM	Provide SAP table name from which you want to read data.

The following table describes the required authorization to write to SAP tables:

Write Object Name	Authorization
S_RFC	/INFADI/GET_TRANSPORT_VERSION, /INFADI/ZPMW, DDIF_FIELDINFO_GET, RFC1, RFCPING, RFC_READ_TABLE
S_TABU_DIS / S_TABU_NUM	Provide SAP table name where you want to write data.

Note: You need to add S_TABU_DIS or S_TABU_NUM based on the version of the SAP system. For more information about S_TABU_DIS or S_TABU_NUM, see the SAP documentation.

Install transport files to read from an SAP table

To read data from SAP tables from a Unicode SAP system, install the SAP Table Reader transport files from the Secure Agent directory to the SAP system.

Ensure that the transport files you install on the SAP machines are the latest. Get the latest transport files from the following directory:

```
<Informatica Secure Agent installation directory>\downloads\package-SAPConnector.<Latest version>\package\rdtm\sap-transport\SAPTableReader
```

Prerequisites to install the transport files

Before you install the SAP Table Reader transports, you need to perform the prerequisite tasks.

- The transport files are applicable for SAP version ECC 5.0 or later.
- Verify that the RSODPABAPCDSVIEW table is available in SAP before you install the TABLE_READER_Addon transport files. If the RSODPABAPCDSVIEW table is not available, the TABLE_READER_Addon transport installation fails.
- Before you install the transports on your production system, install and test the transports in a development system.

The following table lists the transports that you need to install based on the SAP source type that you want to access:

Data and Cofile Names	Transport Request	Functionality
TABLE_READER_R900059.ER6 TABLE_READER_K900059.ER6	ER6K900059	To read data from SAP transparent tables, cluster tables, and pool tables, install only the TABLE_READER transport.
TABLE_READER_Addon_R900085.S4N TABLE_READER_Addon_K900085.S4N	S4NK900085	To read data from ABAP CDS views, install both the TABLE_READER and TABLE_READER_Addon transports. Use the TABLE_READER_Addon transports for SAP NetWeaver 7.50 SP4 version and later. Whenever you install the TABLE_READER transport, you need to reinstall the TABLE_READER_Addon transport even though there is no change in the TABLE_READER_Addon transport version. **Note:** Ensure that you first install the TABLE_READER transport and only then install the TABLE_READER_Addon transport.

Installing transport files

To install the SAP Table Reader transport files, perform the following steps:

1. Find the transport files in the following directory on the Secure Agent machine:
   `<Informatica Secure Agent installation directory>\downloads\package-SAPConnector.<Latest version>\package\rdtm\sap-transport\SAPTableReader`
2. Copy the cofile transport file to the `Cofile` directory in the SAP transport management directory on each SAP machine that you want to access.
   The cofile transport file uses the following naming convention: `TABLE_READER_K<number>.ER6`.

3. Remove "TABLE_READER_" from the file name to rename the cofile.
   For example, for a cofile transport file named `TABLE_READER_K900059.ER6`, rename the file to `K900059.ER6`.
4. Copy the data transport file to the `Data` directory in the SAP transport management directory on each SAP machine that you want to access.
   The data transport file uses the following naming convention: `TABLE_READER_R<number>.ER6`.
5. Remove "TABLE_READER_" from the file name to rename the file.
6. To import the transports to SAP, in the STMS, click **Extras** > **Other Requests** > **Add** and add the transport request to the system queue.
7. In the **Add Transport Request to Import Queue** dialog box, enter the request number for the cofile transport.
   The request number inverts the order of the renamed cofile as follows: ER6K<number>.

   For example, for a cofile transport file renamed as `K900059.ER6`, enter the request number as ER6K900059.
8. In the Request area of the import queue, select the transport request number that you added, and click **Import**.
9. If you want to upgrade from a previous version of the Informatica Transports, select the **Overwrite Originals** option.

Install transport files to write to an SAP table

To write data to SAP custom tables that have been created under the customer namespace, install the SAP Table Writer transport files.

To get and install the latest SAP Table Writer transport files, contact Informatica Global Customer Support.

Connect to SAP table

Let's configure the SAP Table Connector connection properties to connect to SAP tables.

Before you begin

Before you get started, you'll need to configure the Secure Agent machine and SAP system to establish an SAP Table connection.

Check out "Prerequisites" on page 13 to learn more about these tasks.

Connection details

The following table describes the basic connection properties:

Property	Description
Connection Name	Name of the connection. Each connection name must be unique within the organization. Connection names can contain alphanumeric characters, spaces, and the following special characters: _ . + -, Maximum length is 255 characters.
Description	Description of the connection. Maximum length is 4000 characters.
Type	SAP Table Connector
Runtime Environment	The name of the runtime environment where you want to run tasks. Select a Secure Agent or serverless runtime environment. For more information about how to configure a serverless environment, see "Use the serverless runtime environment" on page 27.
Username	The user name with the appropriate user authorization to connect to the SAP account.
Password	The password to connect to the SAP account.
Client	The client number of the SAP application server. Get the required client number from the SAP system to which you want to connect.
Application Server	The host name or IP address of the SAP application server when you read from SAP tables. If you enter the host name or IP address of the SAP application server in this field to read from SAP tables, do not enter the directory of the sapnwrfc.ini file in the **Saprfc.ini Path** field and the DEST entry in **Destination** field. **Note:** This property doesn't apply if you create the connection to write to SAP tables.
System Number	The system number of the SAP application server when you read from SAP tables. If you enter the system number of the SAP application server in this field to read from SAP tables, do not enter the directory of the sapnwrfc.ini file in the **Saprfc.ini Path** field and the DEST entry in **Destination** field. **Note:** This property doesn't apply if you create the connection to write to SAP tables.
Language	Language code that corresponds to the SAP language. Get the required language code from the SAP system to which you want to connect.

Advanced settings

The following table describes the advanced connection properties:

Property	Description

Saprfc.ini Path	The path and file name or the path to the `sapnwrfc.ini` file.
	This property is required if you want to use the connection to write to SAP tables.
	Enter the following directory where the `sapnwrfc.ini` file is available:
	`<Informatica Secure Agent installation directory>/apps/Data_Integration_Server/ext/deploy_to_main/bin/rdtm`
	For the serverless runtime environment, the sapnwrfc.ini file is copied from the AWS location to the following serverless agent directory:
	`/data2/home/cldagnt/SystemAgent/apps/Data_Integration_Server/ext/deploy_to_main/bin/rdtm`
	For more information about how to create the `sapnwrfc.ini` file, see "Configure the sapnwrfc.ini file" on page 20.
	If you enter the directory of the sapnwrfc.ini file in this field, do not enter the host name or IP address, and system number of the SAP application server in the **Application Server** and **System Number** fields.

Destination	DEST entry that you specified in the `sapnwrfc.ini` file for the SAP application server. Use all uppercase letters for the destination. This property is required if you create the connection to write to SAP tables. If you enter the DEST entry in this field, do not enter the host name or IP address, and system number of the SAP application server in the **Application Server** and **System Number** fields.
Port Range	HTTP port range. The SAP Table connection uses the specified port numbers to connect to SAP tables using the HTTP protocol. Default range is 10000-65535. Enter a range in the default range, for example, 10000-20000. When a range is outside the default range, the connection uses the default range.
Test Streaming	Tests the connection. When selected, tests the connection using both RFC and HTTP protocol. When not selected, tests connection using RFC protocol.
Https Connection	Connects to SAP through HTTPS protocol. To successfully connect to SAP through HTTPS, verify that your administrator has configured the Secure Agent machine and the SAP system. For more information about how to connect to SAP through HTTPS, see "Configure HTTPS to connect to SAP" on page 22.
Keystore Location	Absolute path and file name of the keystore file to connect to SAP. Specify both the path and file name in the following format: `<Directory>/<Keystore file name>.jks`
Keystore Password	The destination password to access the keystore file.

Property	Description
Private Key Password	The export password to access the .P12 file.

SAP Additional Parameters	Additional SAP properties that the Secure Agent uses to connect to the SAP system as an RFC client.
	Specify the required RFC-specific parameters and connection information that enable the Secure Agent to connect to SAP.
	For example, you can specify the load balancing parameters listed in the following sample:
	`MSHOST=<Host name of the message server>`
	`R3NAME=<Name of the SAP system>`
	`group=<Group name of the application server>`
	If you configure a parameter in other connection property fields, do not enter the same parameter value in the **SAP Additional Properties** field.
	For more information about RFC-specific parameters, see the SAP documentation.

Configure the sapnwrfc.ini file

To enable the Secure Agent to connect to the SAP system as an RFC client, create and configure the `sapnwrfc.ini` file on the Secure Agent machine.

If you want to write data to SAP tables, you need to use the `sapnwrfc.ini` file.

SAP uses the communications protocol, Remote Function Call (RFC), to communicate with other systems. SAP stores RFC-specific parameters and connection information in a file named `sapnwrfc.ini`.

If you are upgrading from an earlier version, you do not need to create an `sapnwrfc.ini` file. The Secure Agent copies the `sapnwrfc.ini` file to the `deploy_to_main\bin\rdtm` directory.

When you read data from or write data to SAP tables, if you define the path and file name of the `sapnwrfc.ini` file in the SAP connection, the Secure Agent uses the `sapnwrfc.ini` file. However, if you define only the path of the `sapnwrfc.ini` file in the connection, the Secure Agent first verifies if an `sapnwrfc.ini` file exists in the specified path. If the `sapnwrfc.ini` file exists, the Secure Agent uses the `sapnwrfc.ini` file. Else, an exception occurs.

To create the `sapnwrfc.ini` file, use a DOS editor or WordPad to configure the `sapnwrfc.ini` file. Notepad can introduce errors to the `sapnwrfc.ini` file.

After you create the `sapnwrfc.ini` file, copy the file to the following directory and restart the Secure Agent:

```
<Informatica Secure Agent installation directory>\apps\Data_Integration_Server\ext
\deploy_to_main\bin\rdtm\
```

Create the `deploy_to_main\bin\rdtm` directory if it does not already exist.

Using the configured sapnwrfc.ini file in connections

After you create the sapnwrfc.ini file, you can use the `sapnwrfc.ini` file to configure the following types of connections that you want to use:

Connection to a specific SAP application server

Create this connection to enable communication between an RFC client and an SAP system. Each connection entry specifies one application server and one SAP system.

The following sample shows a connection entry for a specific SAP application server in the `sapnwrfc.ini` file:

```
DEST=sapr3
ASHOST=sapr3
SYSNR=00
```

Connection to use SAP load balancing

Create this connection to enable SAP to create an RFC connection to the application server with the least load at run time. Use this connection when you want to use SAP load balancing.

The following sample shows a connection entry for SAP load balancing in the `sapnwrfc.ini` file:

```
DEST=sapr3
R3NAME=ABV
MSHOST=infamessageserver.informatica.com
GROUP=INFADEV
```

Connection to an RFC server program registered at an SAP gateway

Create this connection to connect to an SAP system from which you want to receive outbound IDocs.

The following sample shows a connection entry for an RFC server program registered at an SAP gateway in the `sapnwrfc.ini` file:

```
DEST=sapr346CLSQA
PROGRAM_ID=PID_LSRECEIVE
GWHOST=sapr346c
GWSERV=sapgw00
```

You can configure the following parameters in the `sapnwrfc.ini` file for various connection types:

sapnwrfc.ini Parameter	Description	Applicable Connection Types
DEST	Logical name of the SAP system for the connection. All DEST entries must be unique. You need to have only one DEST entry for each SAP system. For SAP versions 4.6C and later, use up to 32 characters. For earlier versions, use up to eight characters.	Use this parameter for the following types of connections: - Connection to a specific SAP application server - Connection to use load balancing - Connection to an RFC server program registered at an SAP gateway
ASHOST	Host name or IP address of the SAP application. The Secure Agent uses this entry to attach to the application server.	Use this parameter to create a connection to a specific SAP application server.
SYSNR	SAP system number.	Use this parameter to create a connection to a specific SAP application server.
R3NAME	Name of the SAP system.	Use this parameter to create a connection to use SAP load balancing.
MSHOST	Host name of the SAP message server.	Use this parameter to create a connection to use SAP load balancing.
GROUP	Group name of the SAP application server.	Use this parameter to create a connection to use SAP load balancing.

sapnwrfc.ini Parameter	Description	Applicable Connection Types
PROGRAM_ID	Program ID. The Program ID must be the same as the Program ID for the logical system that you define in the SAP system to send or receive IDocs.	Use this parameter to create a connection to an RFC server program registered at an SAP gateway.
GWHOST	Host name of the SAP gateway.	Use this parameter to create a connection to an RFC server program registered at an SAP gateway.
GWSERV	Server name of the SAP gateway.	Use this parameter to create a connection to an RFC server program registered at an SAP gateway.
TRACE	Debugs RFC connection-related problems. Set one of the following values based on the level of detail that you want in the trace: - 0. Off - 1. Brief - 2. Verbose - 3. Full	Use this parameter for the following types of connections: - Connection to a specific SAP application server - Connection to use load balancing - Connection to an RFC server program registered at an SAP gateway

The following snippet shows a sample `sapnwrfc.ini` file:

```
/*=====================================================================*/
/* Connection to an RFC server program registered at an SAP gateway */
/*=====================================================================*/
DEST=<destination in RfcRegisterServer>
PROGRAM_ID=<program-ID, optional; default: destination>
GWHOST=<host name of the SAP gateway>
GWSERV=<service name of the SAP gateway>
*=====================================================================*/
/* Connection to a specific SAP application server */
/*=====================================================================*/
DEST=<destination in RfcOpenConnection>
ASHOST=<Host name of the application server.>
SYSNR=<The back-end system number.>
/*=====================================================================*/
/* Connection to use SAP load balancing */
/* The application server will be determined at run time. */
/*=====================================================================*/
DEST=<destination in RfcOpenConnection>
R3NAME=<name of SAP system, optional; default: destination>
MSHOST=<host name of the message server>
GROUP=<group name of the application servers, optional; default: PUBLIC>
```

Configure HTTPS to connect to SAP

To connect to SAP through HTTPS and read SAP table sources, you need the OpenSSL certificate in the Secure Agent machine and the SAP system.

Create an OpenSSL certificate in the Secure Agent machine. Then, import the created certificate in the PSE format to the SAP system truststore.

To enable HTTPS in an SAP Table connection, you also need to specify the generated keystore password and private key password of the keystore file in the SAP system from transaction code SAP ICM Monitor (SMICM).

Create an OpenSSL certificate

Before you create an OpenSSL certificate, you need to perform the prerequisite tasks.

- Download and install OpenSSL on the Secure Agent machine.
- Based on the operating system of the machine that hosts the Secure Agent and the SAP system, download the latest available patch of the SAPGENPSE Cryptography tool from the SAP Service Marketplace.
  Typically, the SAPGENPSE files are extracted to the `nt-x86_64` directory.
- Configure the following SAP parameters: `icm/server_port`, `ssl/ssl_lib`, `sec/libsapsecu`, `ssf/ssfapi_lib`, `ssf/name`, `icm/HTTPS/verify_client`, `ssl/client_pse`, and `wdisp/ssl_encrypt`.
  For more information, see the SAP documentation.

To create a self-signed certificate using OpenSSL, perform the following tasks:

1. From the command line, set the `OPENSSL_CONF` variable to the absolute path to the `openssl.cfg` file.
   For example, run the following command: `set OPENSSL_CONF= C:\OpenSSL-Win64\bin\openssl.cfg`
2. Navigate to the `<openSSL installation directory>\bin` directory.
3. To generate a 2048-bit RSA private key, run the following command:
   `openssl.exe req -new -newkey rsa:2048 -sha1 -keyout <RSAkey File_Name>.key -out <RSAkey File_Name>.csr`
4. When prompted, enter the following values:
   - Private key password (PEM pass phrase). Enter a phrase that you want to use to encrypt the secret key. Re-enter the password for verification.

     Important: Make a note of this PEM password. You need to keep this password handy while creating a self-signed key and PKCS#12 certificate.
   - Two-letter code for country name.
   - State or province name.
   - Locality name.
   - Organization name
   - Organization unit name.
   - Common name (CN). Mandatory.

     Important: Enter the fully qualified host name of the machine that hosts the Secure Agent.
   - Email address.
5. Optionally, enter the following attributes that you want to pass along with the certificate request:
   - Challenge password.
   - Optional company name.

   A RSA private key of 2048-bit size is created. The `<RSAkey File_Name>.key and <RSAkey File_Name>.csr` files are generated in the specified directory.
6. To generate a self-signed key using the RSA private key, run the following command:
   `openssl x509 -req -days 11499 -in <RSAkey File_Name>.csr -signkey <RSAkey File_Name>.key -out <Certificate File_Name>.crt`
7. When prompted, enter the PEM pass phrase for the RSA private key.
   The `<Certificate File_Name>.crt` file is generated in the specified directory.

8. To concatenate the contents of the `<Certificate File_Name>.crt` file and the `<RSAkey File_Name>.key` file to a `.pem` file, perform the following tasks:
   a. Open the `<Certificate File_Name>.crt` file and the `<RSAkey File_Name>.key` files in a Text editor.
   b. Create a file and save it as `<PEM File_Name>.pem`.
   c. Copy the contents of the `<Certificate File_Name>.crt` file and paste it in the `.pem` file.
   d. Copy the contents of the `<RSAKey_Name>.key` file and append it to the existing contents of the `.pem` file.
   e. Save the `<PEM file name>.pem` file.
9. To create a PKCS#12 certificate, run the following command from the command line:
   `openssl pkcs12 -export -in <PEM File_Name>.pem -out <P12 File_Name>.p12 -name "domain name"`
10. When prompted, enter the following details:
    - The PEM pass phrase for the `.pem` file.
    - An export password for the P12 file. Re-enter the password for verification.

      Important: Make a note of this export password for the P12 file. You need to keep this password handy while creating a Java keystore file to connect to SAP through HTTPS.

    The `<P12 File_Name>.p12` file is generated in the specified directory.
11. To create a Java keystore file, enter the following command:
    `keytool -v -importkeystore -srckeystore <P12 File_Name>.p12 -srcstoretype PKCS12 -destkeystore <JKS File_Name>.jks -deststoretype JKS -srcalias "source alias" -destalias "destination alias"`
12. When prompted, enter the following details:
    - Password for the destination keystore, the JKS file.

      Important: Make a note of this password. You need to keep this password handy while creating an SAP Table connection.
    - Password for the source keystore, the P12 file. Enter the Export password for the P12 file.

    The `<JKS File_Name>.jks` file is generated in the specified directory.

    While enabling HTTPS in an SAP Table connection, specify the name and location of this keystore file. You also need to specify the destination keystore password as the Keystore Password and the source keystore password as the Private Key Password.

Convert an OpenSSL certificate to PSE format

After you create an OpenSSL certificate, you need to convert the OpenSSL certificate to PSE format using the SAPGENPSE tool.

1. From the command line, navigate to the `<SAPGENPSE Extraction Directory>` directory.
2. To generate a PSE file, run the following command:
   `sapgenpse import_p12 -p <PSE_Directory>\<PSE File_Name>.pse <P12 Certificate_Directory>\<P12 File_Name>.p12`
3. When prompted, enter the following details:
   - Password for the P12 file. Enter the Export password for the P12 file.
   - Personal identification number (PIN) to protect the PSE file. Re-enter the PIN for verification.

   The `<PSE File_Name>.pse` file is generated in the specified directory.

4. To generate the certificate based on the PSE format, run the following command:
   `sapgenpse export_own_cert -p <PSE File_Directory>\<PSE File_Name>.pse -o <Certificate_Name>.crt`
5. When prompted, enter the PSE PIN number.
   The `<Certificate_Name>.crt` file is generated in the specified directory. Import this certificate file to the SAP system trust store.

Enable the HTTPS service on the SAP system

To configure HTTPS to connect to an SAP system, you need to enable the HTTPS service from the transaction code SAP ICM Monitor (SMICM) in the SAP system

For more information about how to enable the HTTPS service on the SAP system, see the SAP documentation.

Import the certificate to the SAP system trust store

You need to import the certificate in PSE format to the SAP system trust store to connect to SAP through HTTPS.

1. Log in to SAP and go to the STRUST transaction.
2. Select SSL Client (Standard) and specify the password.
   In the **Import Certificate** dialog, you need to select Base64 format as the certificate file format.
3. Click the **Import** icon, and select the `<Certificate_Name>.crt` file in PSE format.

   Note: If a user is on a different SAP network, you might need to add a DNS entry of the agent host on the SAP app server.
4. Click **Add to Certificate List**.
5. Restart the Internet Communication Manager (ICM).

Configure the Secure Network Communication protocol

You can use the SAP Table Connector connection with the Secure Network Communication Protocol to securely read from or write to SAP.

For more information about the steps to configure an SAP SNC connection, see Configure the SAP Secure Network Communication protocol Informatica How-To Library article.

Enable the Secure Agent to operate as a whitelisted host in SAP (Optional)

You can enable the Secure Agent to operate as a whitelisted host when you read SAP table data. Before you enable the Secure Agent as a whitelisted host, verify that the latest transport files are installed.

1. To configure the **JVMOption** property in Administrator to define the Secure Agent as a host that you can add in the HTTP_Whitelist table of the SAP system, perform the following steps:
   a. Select **Administrator** > **Runtime Environments**.
   b. On the **Runtime Environments** page, select the Secure Agent machine that runs the mapping.
   c. Click **Edit**.
   d. In the **System Configuration Details** section, from the **Service** list, select **Data Integration Server**.
   e. Edit any **JVMOption** field to add the following value:
   `-Dsap_whitelist_check=1`
   f. Click **Save**.
   g. Repeat steps b through f for every Secure Agent that you want to define as a host in SAP.
2. Create an entry for the Secure Agent in the SAP HTTP_Whitelist table using the transaction SE16. To create an entry for the Secure Agent, perform the following steps in the SAP system:
   a. Go to transaction SE16.
   b. Configure properties to define the Secure Agent as a host in SAP.
   The following table describes the properties that you need to configure:

Property	Description
MANDT	Required. SAP client number.
ENTRY TYPE	Required. URL type to be compared with this entry. Enter **01** to indicate that the URL is a CSS theme URL.
SORT KEY	Required. Unique value to be used as the primary key. You can enter numbers and alphabets.
PROTOCOL	Protocol that SAP must validate. Enter **HTTP** or **HTTPS**. If you do not enter a value, SAP does not validate the protocol.
HOST	Host machine that SAP must validate. Enter the IP address of the machine that hosts the Secure Agent.
PORT	Port number that SAP must validate. Leave the **Port** field blank to indicate that SAP does not need to validate the port.
URL	URL that SAP must validate. Enter ***** to indicate that SAP does not need to validate the URL.

   Note: If you do not perform step 2, mappings and tasks that run on SAP fail.

3. Repeat steps 1 and 2 for every Secure Agent that you want to configure as a whitelisted host in SAP.

Use the serverless runtime environment

You can use the serverless runtime environment to connect to the SAP system when you configure an SAP Table Connector connection on Linux.

You cannot create an SNC connection when you use the serverless runtime environment.

Before you configure an SAP Table Connector connection using the serverless runtime environment, perform the following tasks:

- Add the libraries in the Amazon S3 bucket or Azure container in your AWS or Azure account.
- Configure the .yml serverless configuration file.
- Configure the JAVA_LIBS property for the serverless runtime environment on Linux.

Add the libraries in the Amazon S3 bucket or Azure container in your AWS or Azure account

Perform the following steps to configure an SAP Table Connector connection in a serverless runtime environment:

1. Create the following structure for the serverless agent configuration in AWS or Azure: `<Supplementary file location>/serverless_agent_config`
2. Add the libraries in the Amazon S3 bucket or Azure container in the following location in your AWS or Azure account: `<Supplementary file location>/serverless_agent_config/sap`

Configure the .yml serverless configuration file

Perform the following steps to configure the .yml serverless configuration file in the serverless runtime environment, and to copy the libraries to the serverless agent directory:

1. Copy the following code snippet to a text editor:

```
version: 1
agent:
  dataIntegrationServer:
    autoDeploy:
      sap:
        jcos:
          - fileCopy:
              sourcePath: sap/jco/<sapjco_libary_filename>
          - fileCopy:
              sourcePath: sap/jco/<sapjco_libary_filename>
        nwrfcs:
          - fileCopy:
              sourcePath: sap/nwrfc/<rfc_libary_filename>
          - fileCopy:
              sourcePath: sap/nwrfc/<sapnwrfc_filename>
```

   where the source path is the directory path of the library files in AWS or Azure.

2. Ensure that the syntax and indentations are valid, and then save the file as `serverlessUserAgentConfig.yml` in the following AWS or Azure location: `<Supplementary file location>/serverless_agent_config`
   When the .yml file runs, the libraries are copied from the AWS or Azure location to the serverless agent directory.

Configure the JAVA_LIBS property for the serverless runtime environment

Perform the following steps in Administrator to configure the JAVA_LIBS property for the serverless runtime environment on Linux:

1. Log in to Informatica Intelligent Cloud Services.
2. Select **Administrator** > **Serverless Environments**.

3. On the **Serverless Environments** tab, expand the Actions menu for the required serverless runtime environment, and then select **Edit**.
4. On the **Runtime Configuration Properties** tab, select **Data Integration Server** as the service and **Tomcat_JRE** as the type.
5. Click **Add Property**.
6. Enter JAVA_LIBS in the **Name** field and set the following value:
   `../bin/rdtm-extra/tpl/sap/sapjco3.jar:../bin/rdtm/javalib/sap/sap-adapter-common.jar`
7. Click **Save**.

For more information about how to configure and use the serverless environment, see "Serverless runtime environment setup" in *Runtime Environments* in the Administrator help.

Troubleshooting an SAP Table connection

The following error displays when I test an SAP Table connection:

```
Test Connection Failed for <connection name>/sap/conn/jco/JCoException
```

Verify that the sapjco3.jar is saved to the following directory:

```
<Informatica Secure Agent installation directory>\apps\Data_Integration_Server\ext
\deploy_to_main\bin\rdtm-extra\tpl\sap
```

Restart the Secure Agent after you copy the sapjco3.jar.

The following error displays when I test an SAP Table connection or use the connection in a task:

```
Test Connection Failed for <connection name>. Error getting the version of the native
layer: java.lang.UnsatisfiedLinkError: no sapjco3 in java.library.path.
```

Verify that the location of the sapjco3.dll file is in the to PATH variable for the Secure Agent machine.

The following error displays when I test an SAP Table connection or use the connection in a task:

```
Test Connection Failed for <connection name>. Error getting the version of the native
layer: java.lang.UnsatisfiedLinkError: no sapjco3 in java.library.path.
```

Add the location of sapjco3.dll to PATH variable and restart the Secure Agent.

A task that reads from SAP tables fails with the following error:

```
Error occurred processing data from SAP : Unable to establish Http Communication between
SAP server and agent! Shutting down reader.
```

The HTTP port is not open or the incoming request is blocked by Windows Firewall. To resolve the issue, in Windows Firewall, use the advanced settings to create a new incoming rule. Apply the rule to TCP and all ports, and choose the HTTP-In protocol.

CHAPTER 3

Mappings and mapping tasks with SAP Table

Use a mapping to define data flow logic that is not available in synchronization tasks, such as specific ordering of logic or joining sources from different systems. Use the Data Integration Mapping Designer to configure mappings.

In advanced mode, the Mapping Designer updates the mapping canvas to include transformations and functions that enable advanced functionality.

When you configure a mapping to describe the flow of data from source and target, you can also add transformations to transform data. A transformation includes field rules to define incoming fields. Links visually represent how data moves through the data flow.

After you create a mapping, you can run the mapping or you can deploy the mapping in a mapping task. The Mapping Configuration application allows you to process data based on the data flow logic defined in a mapping or integration template.

Use the Mapping task wizard to create a mapping task. When you create a mapping task, you select the mapping or integration template for the task to use.

If you configured parameters, which are placeholders for information, in a mapping, you can define the parameters in the mapping task. Defining parameters provides additional flexibility and allows you to use the same mapping in multiple mapping tasks. For example, you can use a parameter for a source connection in a mapping, and then define the source connection when you configure the mapping task.

When you create a mapping task, you can associate the task with a schedule to run it at specified times or on regular intervals. Or, you can run it manually. You can also configure advanced session properties. You can monitor tasks that are currently running and view details about completed tasks.

SAP Table sources in mappings

To read data from an SAP application, configure an SAP table object as the Source transformation in a mapping.

Specify the name and description of the SAP table source. Configure the source and advanced properties for the source object.

The following table describes the source properties that you can configure in a Source transformation:

Property	Description
Connection	Name of the source connection.
Source Type	Select one of the following types: - Single. Select to specify a single SAP Table object. - Multiple. Select to specify multiple SAP Table objects. You can use custom relationships to join multiple source objects. When you create a custom relationship for SAP Table objects, you can select the type of join and the source fields to use. - Parameter. Select to specify a parameter name. You can configure the source object in a mapping task associated with a mapping that uses this source transformation.
Object	Source object. If you specify multiple source objects, you need to create relationships between the source objects.

The following table describes the SAP Table advanced source properties:

Property	Description
Number of rows to be fetched	The number of rows that are randomly retrieved from the SAP Table. Default value of zero retrieves all the rows in the table.
Number of rows to be skipped	The number of rows to be skipped.
Packet size in MB	The HTTP packet size. When you use bulk mode to read data from an SAP table, you can tune the packet size to increase the throughput. Tune the packet size according to the network bandwidth, memory, and CPU resources available on the Secure Agent. Based on the packet size that you configure and the row length, the Secure Agent calculates the number of rows to be read in a single packet. If you increase the packet size, increase the heap size accordingly to improve the throughput. Default is 10 MB.
Data Extraction Mode	You can use one of the following modes to read data from an SAP Table: - Normal Mode. Use this mode to read small volumes of data from the SAP Table. - Bulk Mode. Use this mode to read large volumes of data from the SAP Table. Use bulk mode for better performance. Bulk mode consumes more resources as compared to normal mode. You might need to tune the packet size according to the available resources and data set to increase the performance. Default is normal mode.
Enable Compression	Enables compression. If the Secure Agent and the SAP system are not located in the same network, you might want to enable the compression option to optimize performance.

Property	Description
Update Mode	When you read data from SAP tables, you can configure a mapping to perform delta extraction. You can use one of the following options based on the update mode that you want to use: - 0- Full. Use this option when you want to extract all the records from an SAP table instead of reading only the changed data. - 1- Delta initialization without transfer. Use this option when you do not want to extract any data but want to record the latest change number in the Informatica custom table /INFADI/TBLCHNGN for subsequent delta extractions. - 2- Delta initialization with transfer. Use this option when you want to extract all the records from an SAP table to build an initial set of the data and subsequently run a delta update session to capture the changed data. - 3- Delta update. Use this option when you want to read only the data that changed since the last data extraction. - 4- Delta repeat. Use this option if you encountered errors in a previous delta update and want to repeat the delta update. - Parameter. When you use this option, the Secure Agent uses the update mode value from a parameter file. Default is 0- Full.
Parameter Name for Update Mode	The parameter name that you defined for update mode in the parameter file.
Override Table Name for Delta Extraction	Overrides the SAP table name with the SAP structure name from which you want to extract delta records that are captured with the structure name in the CDPOS table.
Advanced Properties	Advanced properties for the SAP Table object to run mappings. If you specify more than one property, separate each property-value pair with a semicolon in the following format: `<Property name1>=<Property value1>;<Property name2>=<Property value2>` For more information about the advanced properties, see "Advanced properties" on page 36.
Tracing Level	Sets the amount of detail that appears in the log file. Select one of the following tracing level options from the list: - Terse - Normal - Verbose Initialization - Verbose Data Default is Normal.

Filter options

You can configure the Source transformation to filter data before the data enters the data flow. Use the source query options to filter source data.

To filter data, configure the source query options on the **Source** tab of the Source transformation. Expand the **Query Options** section, and configure the filter condition.

When you configure a filter, you can use either a simple or advanced filter. You can also use a parameter in a filter expression and define the filter expression in the task.

You can configure the following filters in the Source transformation:

Simple data filter

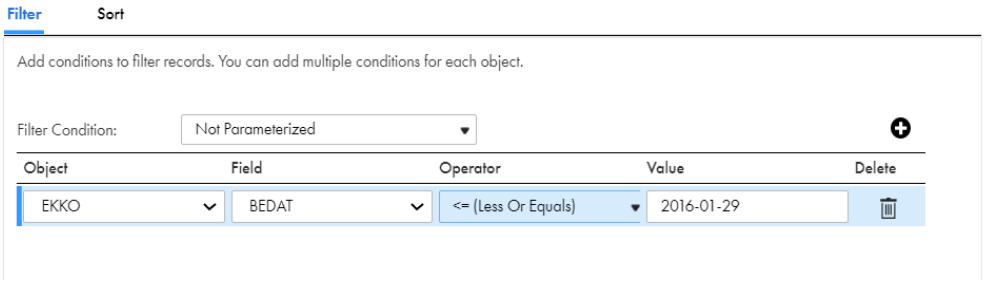
To use a simple data filter, select a source object, source field, operator, and then enter the value.

For example, to filter data from the BEDAT field in the EKKO table when the date is less than or equal to 2016-01-29, use the following format: `EKKO BEDAT <= 2016-01-29`

You can also use a parameter in a filter expression in a simple data filter.

For example, `EKKO EBELN = $$PARAM`.

The following image shows the configured simple data filter that filters data from the BEDAT field in the EKKO table when the date is less than or equal to 2016-01-29:

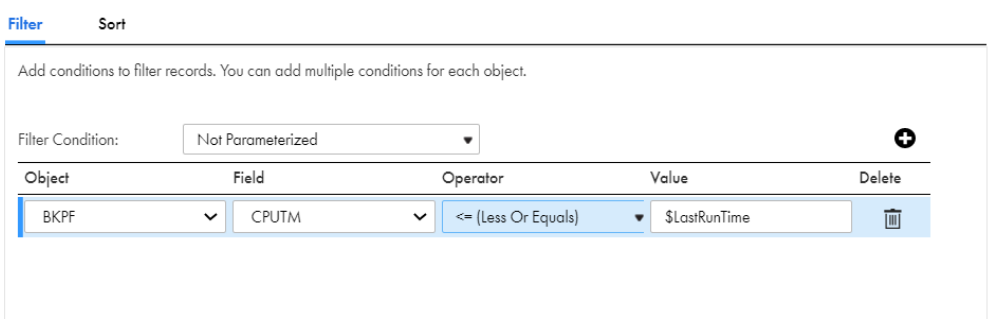


Simple data filter using $LastRunTime variable

To use the $LastRunTime variable in a simple data filter, select a source object, source field, operator, and then enter the value.

For example, to filter data from the CPUTM field in the BKPF table when the data is less than or equal to the LastRunTime variable, use the following format: `BKPF CPUTM <= $LastRunTime`

The following image shows the configured simple data filter that filters data from the CPUTM field in the BKPF table when the data is less than or equal to the LastRunTime variable:



Simple data filter in an ABAP CDS view object

To use a simple data filter in an ABAP CDS view object, select an ABAP CDS view source object, source field, operator, and then enter the value.

For example, to filter data from the paramO_P2 field in the ZSAN_CDS_OPT_PARAM ABAP CDS view object when the value is not equal to 10, use the following format:

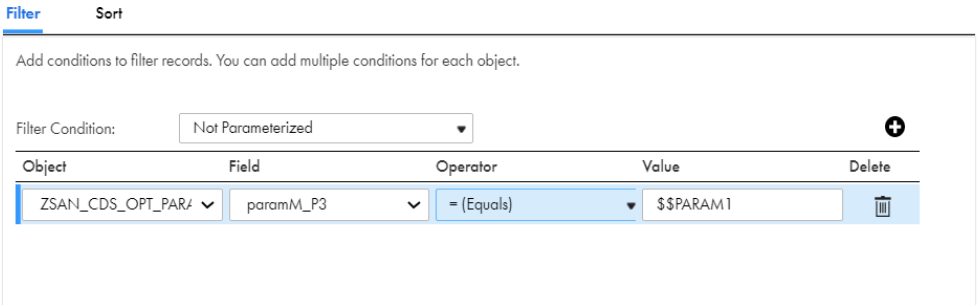
```
ZSAN_CDS_OPT_PARAM paramO_P2 <> 10
```

You can also use a parameter in a filter expression in a simple data filter to filter data from an ABAP CDS view object.

For example, `ZSAN_CDS_OPT_PARAM paramM_P3 = $$PARAM1`.

In the example, paramO denotes an optional parameter and paramM denotes a mandatory parameter.

The following image shows the configured simple data filter that filters data from the paramM_P3 field in an ABAP CDS view object when the data matches with the defined parameter:



Advanced data filter using single condition

To use an advanced data filter that contains a single condition, select **Advanced** as the type of filter, and then enter the field expression in the following format:

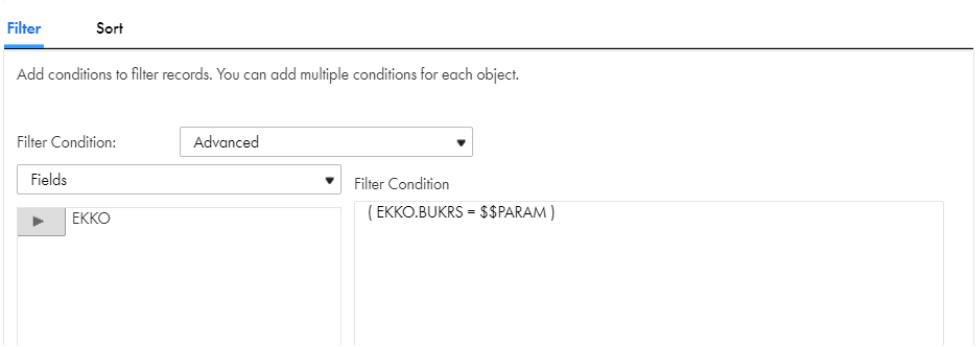
```
( <TableName.Field> <Operator> <'Value'> )
```

For example, to filter data from the BUKRS field in the EKKO table when the value is 1010, use the following format: `( EKKO.BUKRS = '1010' )`

You can also use a parameter in a filter expression in an advanced data filter.

For example, `( EKKO EBELN = $$PARAM ).`

The following image shows the configured advanced data filter that filters data of the BUKRS field from the EKKO table when the data matches with the defined parameter:



Advanced data filter using multiple conditions

To use an advanced data filter that contains multiple conditions, select **Advanced** as the type of filter, and then enter the field expression in the following format:

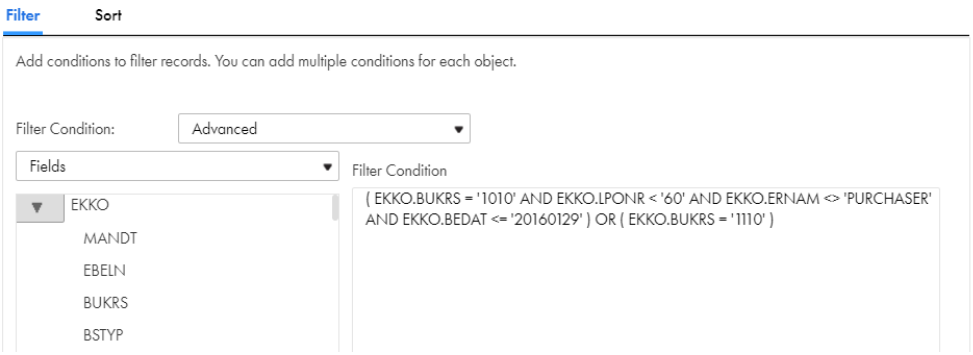
```
( <TableName.Field> <Operator> <'Value'> AND <Table name.Field> <Operator> <'Value'> ) OR
( <Table name.Field> <Operator> <'Value'> AND <Table name.Field> <Operator> <'Value'> )
```

For example, to filter data from multiple fields in the EKKO table, use the following format that contains a logical expression: `( EKKO.BUKRS = '1010' AND EKKO.LPONR < '60' AND EKKO.ERNAM <> 'PURCHASER' AND EKKO.BEDAT <= '20160129' ) OR ( EKKO.BUKRS = '1110' )`

You can also use a parameter in a filter expression in an advanced data filter.

For example, `( EKKO.BUKRS = $$PARAM AND EKKO.LPONR < $$PARAM1 AND EKKO.ERNAM <> 'PURCHASER' AND EKKO.BEDAT <= $$PARAM2 ) OR ( EKKO.BUKRS = $$PARAM3 ).`

The following image shows the configured advanced data filter that filters data from multiple fields in the EKKO table using the expression that contains the AND and OR logical conditions:



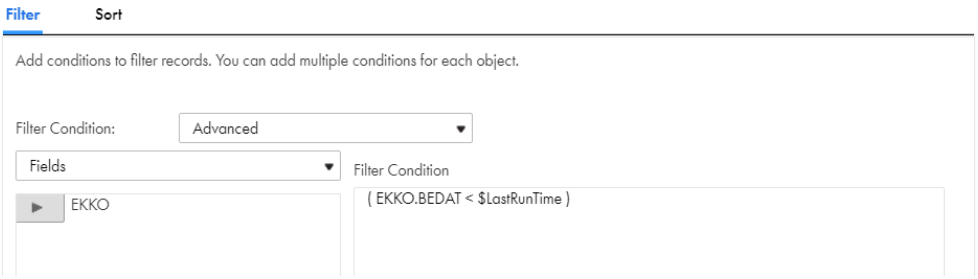
Advanced data filter using $LastRunTime variable

To use the $LastRunTime variable in an advanced data filter, select **Advanced** as the type of filter, and then enter the field expression in the following format:

```
( <Table name.Field> <operator> <Value> )
```

For example, to filter data from the BEDAT field in the EKKO table when the data is less than the LastRunTime variable, use the following format: `( EKKO.BEDAT < $LastRunTime )`

The following image shows the configured advanced data filter that filters data from the BEDAT field in the EKKO table when the data is less than the LastRunTime variable:



Advanced data filter using SY-DATUM variable

To use the SY-DATUM variable in an advanced data filter, select **Advanced** as the type of filter, and then enter the field expression in the following format based on the transports you installed:

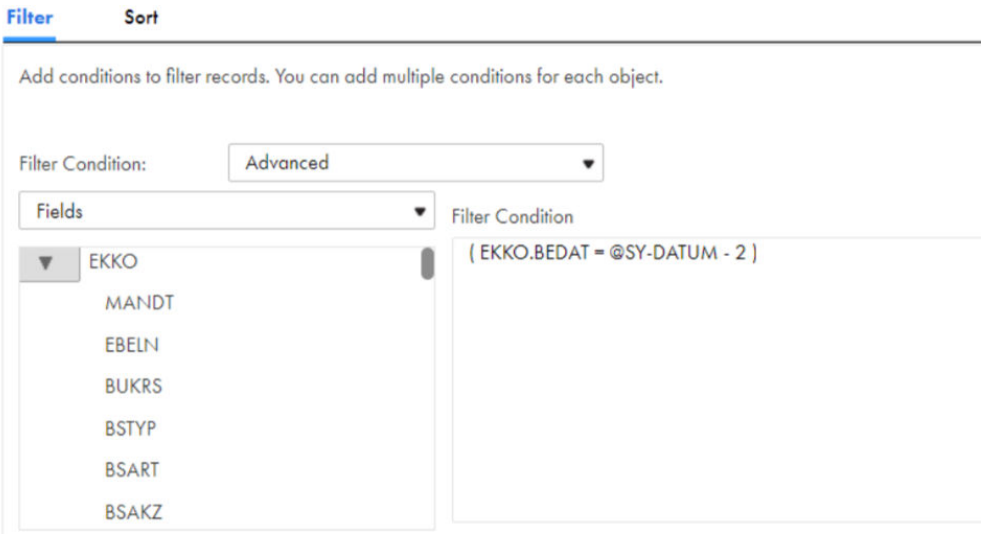
```
( <Table name.Field> <operator> <Value> )
```

For example, to filter data from the BEDAT field in the EKKO table when the data is two days older than the current date and you installed the TABLE_READER transport, use the following format: `( EKKO.BEDAT = SY-DATUM - 2 )`

You can also use the SY-DATUM variable in an advanced data filter when you installed the TABLE_READER_Addon transport.

For example, use the following format: `( EKKO.BEDAT = @SY-DATUM - 2 )`

The following image shows the configured advanced data filter that filters data from the BEDAT field in the EKKO table when the data is two days older and you installed the TABLE_READER_Addon transport:



Sort options

You can configure the Source transformation to sort data before the data enters the data flow. Use the source query options to sort source data.

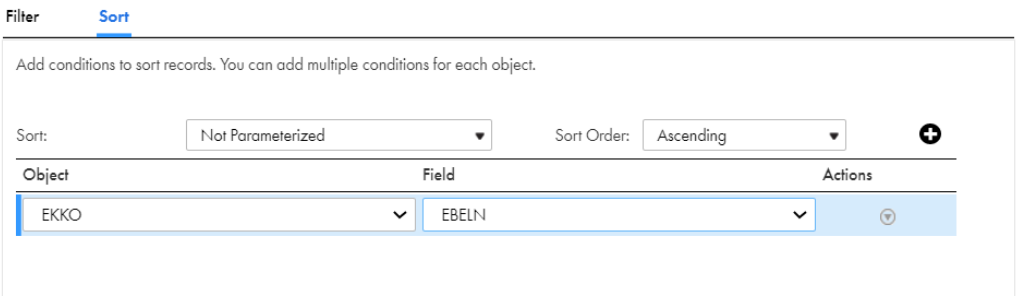
To sort data, configure the source query options on the **Source** tab of the Source transformation. Expand the **Query Options** section, and configure the sort condition.

When you configure a sort condition, you can sort data either in the ascending or descending order from the field in a table.

To use the sort condition, select the source object, sort by field, and then the sort direction.

For example, to sort data in the ascending order from the EBELN field in the EKKO table, use the following format: `EKKO EBELN Ascending`

The following image shows the configured sort condition that sorts data of the EBELN field from the EKKO table in ascending order:



Join conditions

You can configure a relationship between the selected source object and related source object.

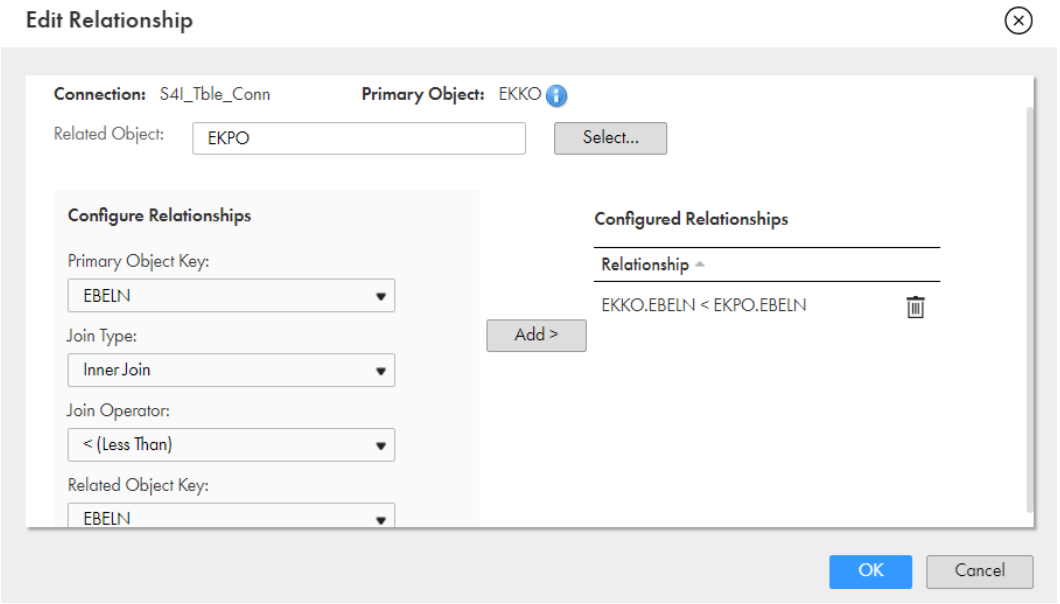
When you select multiple source objects as the source type, you can configure a relationship to join multiple source objects.

To configure a relationship using the join condition between the source and the related objects, specify the key field in the source SAP object, the type of join, the join operator, the related SAP object and the key field in the related object, and then click **ADD**.

For example, to configure an inner join to join the EKKO and EKPO tables when the value of the EBELN field of the EKPO table is less than the value of the EBELN field of the EKKO table, use the following format:

```
EKKO.EBELN Inner Join < EKPO.EBELN
```

The following image shows a configured custom relationship that uses an inner join to join the EKKO and EKPO tables when the value of the EBELN field of the EKPO table is less than the value of the EBELN field of the EKKO table:



Advanced properties

You can configure additional options in the Advanced Properties field in the Source transformation.

You can use the following properties as advanced properties when you configure delta extraction on SAP source tables:

- When the external application time zone differs from the SAP system time zone, enter the SAP system offset time in minutes in the following format:

  ```
  delta_offset=<SAP system offset time in minutes>
  ```

  For example, if the difference between the external application time zone and SAP system time zone is 480 minutes, enter the following value:

  ```
  delta_offset=480
  ```

- To fetch the changed data of key fields that are marked for hard deletion in SAP to the target table, enter the following advanced property:

  ```
  fetch_del_rows=true
  ```

You can configure the `fetch_del_rows=true` advanced property in delta extraction mappings using the following guidelines:

- To fetch the deleted delta records to the target table, select **Data Driven** as the operation and **Update Else Insert** as the update mode in the Target transformation. Otherwise, data corruption occurs in the target.
- When you perform delta extraction for the deleted records, the Secure Agent fetches only the key value for the deleted delta records.

Delta extraction for SAP table reader mappings

When you read data from SAP tables, you can configure a mapping to perform delta extraction. With delta extraction, you can choose to read only the changed data.

The SAP table and SAP columns for which you want to perform delta extraction must be part of a change document object in SAP. For more information about creating a change document in SAP, see the SAP documentation.

The Secure Agent uses the CDHDR (Change Document Header) and CDPOS (Change Document Position) tables in SAP to extract the changed data. The CDHDR table stores the change document header information. The CDPOS table stores the new value and the old value of the changed data. The Secure Agent uses the change document number in the CDHDR table to find the latest change number in the CDPOS table.

Delta extraction behavior

When you configure a delta extraction, the Secure Agent does not fetch the change indicators marked for insert, delete, or update for delta records from SAP. Hence, the delta rows that are extracted from the source are marked for upsert by default.

When you perform a delta extraction and if the row is available in the SAP target table, the Secure Agent updates the delta records to the target table. If the row is not available, the Secure Agent inserts the records to the SAP target table.

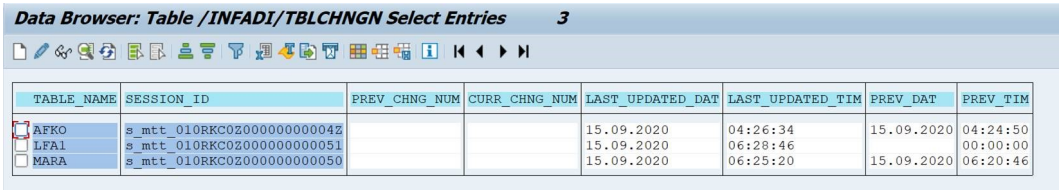
If multiple transactions such as insert, update, or delete occurs for the same record in the SAP source, the Secure Agent fetches only one record. However, if an operation, for example, an insert for a record occurs in the SAP source table and you run the mapping, the Secure Agent fetches the inserted record. Later, if an update occurs for the same record in the SAP source table and you run the delta extraction mapping, the Secure Agent fetches the updated record.

When you perform a delta extraction for the deleted records in the SAP table, the Secure Agent fetches only the key value of the deleted delta records to the target table.

Informatica custom table /INFADI/TBLCHNGN

The Secure Agent creates and maintains an Informatica custom table in SAP named /INFADI/TBLCHNGN. The /INFADI/TBLCHNGN table captures the time that is used for delta extraction through the Informatica mappings. You can use transaction SE11 or SE16 to view the table entries.

The following image shows the /INFADI/TBLCHNGN table:



TABLE_NAME	SESSION_ID	PREV_CHNG_NUM	CURR_CHNG_NUM	LAST_UPDATED_DAT	LAST_UPDATED_TIM	PREV_DAT	PREV_TIM
AFKO	s_mtt_010RKC0Z000000000004Z			15.09.2020	04:26:34	15.09.2020	04:24:50
LFA1	s_mtt_010RKC0Z000000000051			15.09.2020	06:28:46		00:00:00
MARA	s_mtt_010RKC0Z000000000050			15.09.2020	06:25:20	15.09.2020	06:20:46

The /INFADI/TBLCHNGN table contains the following columns:

TABLE_NAME

Specifies the SAP source table name from which data is extracted.

SESSION_ID

Specifies the unique Informatica session ID for delta extraction. The Secure Agent generates a unique session ID for each mapping run for a particular SAP table.

PREV_CHNG_NUM

This column does not apply in delta extraction mappings.

CURR_CHNG_NUM

This column does not apply in delta extraction mappings.

PREV_DAT

Indicates the date from when changes were extracted in the last delta extraction. The Secure Agent uses this information when you use the Delta Repeat option.

PREV_TIME

Indicates the time from when the changes were extracted in the last delta extraction. The Secure Agent uses this information when you use the Delta Repeat option.

LAST_UPDATED_DAT

Indicates the date up to when the changes were extracted from the source in the last delta extraction. This value also indicates the date from when the changes will be extracted for the subsequent delta extraction.

LAST_UPDATED_TIM

Indicates the time up to when the changes were extracted from the source in the last delta extraction. This value also indicates the time from when the changes will be extracted for the subsequent delta extraction.

Update modes for delta extraction

When you configure delta extraction for SAP table reader mappings, you can select the update mode that you want to use.

You can select one of the following update modes:

Full

If you select the **Full** option, the Secure Agent extracts all the records from an SAP table. The Secure Agent does not update any details in the Informatica custom table /INFADI/TBLCHNGN.

Use this option when you want to extract all the records from an SAP table instead of reading only the changed data.

Default is **Full**.

Delta initialization without transfer

If you select the **Delta initialization without transfer** option, the Secure Agent does not extract any data from an SAP table but records the LAST_UPDATED_DAT and LAST_UPDATED_TIM in the Informatica custom table `/INFADI/TBLCHNGN` for subsequent delta extractions.

The Secure Agent performs the following actions:

- Sets the values for LAST_UPDATED_DAT and LAST_UPDATED_TIM columns.
- Initializes and sets the values for the PREV_DAT and PREV_TIM columns.

Use this option when you do not want to extract any data but you want to record the LAST_UPDATED_DAT and LAST_UPDATED_TIM in the Informatica custom table /INFADI/TBLCHNGN for subsequent delta extractions.

For example, you have a table named **Customers** that contains 5 million records. You want to read the initial set of records by using another product such as Informatica Data Replication and then write the records to a Teradata data warehouse. You then want to use SAP Connector to read only the new customer records that get added to the table. In this case, you can configure delta initialization without transfer and then subsequently run a delta update session to capture the changed data.

Delta initialization with transfer

If you select the **Delta initialization with transfer** option, the Secure Agent extracts all the records from an SAP table but you want to record the LAST_UPDATED_DAT and LAST_UPDATED_TIM in the Informatica custom table /INFADI/TBLCHNGN for subsequent delta extractions.

The Secure Agent performs the following actions:

- Sets the LAST_UPDATED_DAT and LAST_UPDATED_TIM columns.
- Initializes the PREV_DAT and PREV_TIM.
- Extracts all the data present in the SAP table.

Use this option when you want to extract all the records from an SAP table to build an initial set of the data and subsequently run a delta update session to capture the changed data.

Delta update

If you select the **Delta update** option, the Secure Agent extracts the changed data since the last data extraction.

The Secure Agent performs the following actions:

- Extracts records from the columns LAST_UPDATED_DAT AND LAST_UPDATED_TIM to the current date and time.
- Moves the values from columns LAST_UPDATED_DAT and LAST_UPDATED_TIM to PREV_DAT and PREV_TIM, respectively.
- Updates the values in columns LAST_UPDATED_DAT and LAST_UPDATED_TIM to the current date and time.

Use this option when you want to read only the data that changed since the last data extraction.

Before you run a delta update session, you need to perform a delta initialization with transfer or delta initialization without transfer. The delta initialization records the LAST_UPDATED_DAT AND LAST_UPDATED_TIM that the Secure Agent uses to run a delta update session.

Note: To avoid data loss, the current date and time is frozen before the Secure Agent runs the query. If any data enters at the time when the mapping runs, that data is extracted only when you run the next mapping.

Delta repeat

If you select the **Delta repeat** option, the Secure Agent repeats a previous delta update in case of errors. It returns records from the PREV_DAT and PREV_TIM values in the Informatica custom table /INFADI/TBLCHNGN to the LAST_UPDATED_DAT and LAST_UPDATED_TIM values in the Informatica custom table /INFADI/TBLCHNGN. It does not update any change numbers in the /INFADI/TBLCHNGN table.

Use this option if you encountered errors in a previous delta update and want to repeat the delta update.

Before you run a delta repeat session, you need to perform a delta update.

Parameter

If you select the **Parameter** option, the Secure Agent uses the update mode value from a parameter file. Define a parameter name and parameter value in a parameter file. In the SAP table reader mapping that you create for delta extraction, specify the same parameter name that you defined in the parameter file. Then, create a mapping task and specify the parameter file name in the task. Instead of updating the parameter value in the mapping every time, you can update the parameter value in the parameter file and run the mapping task again.

Configuring a parameter file for delta extraction

A parameter file is a list of user-defined parameters and their associated values. To perform a delta extraction, you can specify the update mode in a parameter file so that you do not need to edit the mapping every time you want to change the update mode.

To use a parameter file, perform the following steps:

1. Create a parameter file in the following directory:

   `<Secure Agent installation directory>/apps/Data_Integration_Server/data/userparameters`
2. In the parameter file, enter a parameter name and specify the parameter value that you want to use.

   The parameter name must start with `$$` and cannot contain space characters.

   You can use one of the following parameter values based on the update mode that you want to use:

   - **0**. Use for Full.
   - **1**. Use for Delta initialization without transfer.
   - **2**. Use for Delta initialization with transfer.
   - **3**. Use for Delta update.
   - **4**. Use for Delta repeat.

   Use the following format to specify the parameter name and parameter value:

   `$$<parameter_name>=<parameter_value>`

   Do not use space characters while specifying the parameter name and parameter value.

   For example, enter: `$$deltaparameter=0`
3. Save the parameter file.
4. Open the SAP table reader mapping that you want to use for delta extraction.
5. Select **Parameter** from the **Update Mode** list.

6. In the **Parameter Name for Update Mode** field, enter the parameter name that you defined in the parameter file.
7. Create a mapping task based on the SAP table reader mapping.
8. In the **Schedule** page of the mapping task, enter the parameter file name under the **Advanced Options** section.
9. Run the mapping task.

   To change the update mode, update the parameter value in the parameter file, and run the mapping task again. For example, the first time you run a mapping task, you can specify the parameter value as **2** to use delta initialization with transfer. After the initial extraction is done, you might want to change the parameter value to **3** to capture only the changed data in the next mapping task run. Instead of updating the parameter value in the mapping every time, you can update the parameter value in the parameter file and run the mapping task again.

Configure a table name override for delta extraction

When you configure a SAP Table reader mapping, you can override the selected table name at runtime with the structure name to perform a delta extraction.

When you run an SAP Table reader mapping for delta extraction, the Secure Agent fetches changed records from the SAP table for entries logged for the SAP table name in the Change Document Position (CDPOS) table. If there are entries logged for the SAP table structure in the CDPOS table, you can fetch those records by overriding the table name in the Override Table Name for Delta Extraction field in the SAP Table advanced source properties.

For example, if the table name you specified as the object type in the mapping is *CRMD_ORDERADM_H*, to get the delta records for entries captured in the CDPOS table for the structure name, specify the structure name *CRMA_ORDERADM_H* in the **Override Table Name for Delta Extraction** field. The Secure Agent fetches records from *CRMA_ORDERADM_H* that has change entries logged for the table structure.

If the delta data captured in the CDPOS table does not include the structure name, keep this field blank.

Rules and guidelines for delta extraction

Consider the following rules and guidelines when you configure delta extraction for SAP table reader mappings:

- You can perform delta extraction only for a single SAP table. You can't use native joins to join data from two or more SAP tables.
- You can't configure delta extraction to look up data from SAP tables.
- You can't configure delta extraction with partitioning.
- You can't configure delta extraction for multiple pipelines within the same mapping.
- When the Secure Agent performs a delta extraction, it does not retrieve the records in the same order in which they were inserted or updated in the SAP table. For example, record 10 might have been updated first in the SAP table before record 20. However, while extracting the data, the Secure Agent might fetch record 20 first and then record 10.
- The Secure Agent does not print any information in the session log to indicate whether the records extracted through delta extraction were part of an insert or update operation in SAP.
- During delta extraction, if there are multiple entries for a key between the current and last updated date and time, the Secure Agent fetches only the latest entry for the key.

For example, consider that a record was inserted into a Customer table in SAP with the customer name set to `John`. The record was later updated and the name was changed to `Bill`. The Secure Agent fetches the name value as only `Bill`.

- You can't use the QUAN and CURR data types in delta extraction mappings.
- You can't configure delta extraction in data transfer tasks.
- To fetch the inserted or updated delta records to the target table, select **Upsert** as the operation and **Update Else Insert** as the update mode in the Target transformation. Otherwise, data corruption occurs in the target.

Configuring delta extraction for an SAP table reader mapping

To configure delta extraction for an SAP table reader mapping, select the update mode that you want to use and optionally define a parameter in the mapping.

1. Create a mapping to read data from an SAP table and write data to a target.
2. Click the Source transformation in the mapping.
3. In the **Properties** panel, click the **Source** tab.
4. Under the advanced properties, select one of the following values from the **Update Mode** list:
   - **0 - Full**
   - **1 - Delta initialization without transfer**
   - **2 - Delta initialization with transfer**
   - **3 - Delta update**
   - **4 - Delta repeat**
   - **Parameter**

   For information about using a parameter for delta extraction, see "Configuring a parameter file for delta extraction" on page 40.
5. Save and run the mapping.

Troubleshooting delta extraction for SAP Table Reader mappings

Why do I see the error "Only Full Update Mode is supported for table {table_name} because it is not a part of any change document object in SAP"?

The error occurs because the SAP table for which you are trying to perform delta extraction is not part of a change document object in SAP.

If the SAP table and SAP columns for which you want to perform delta extraction are not part of a change document object in SAP, you cannot perform delta extraction. You can only perform a full extraction.

Why do I see the error "An error occurred while fetching the current date and time from SAP because there is no entry present in the /INFADI/TBLCHNGN table. Run a delta initialization session first."?

The error occurs when you run a delta update or delta repeat session directly without performing a delta initialization.

The delta initialization records the LAST_UPDATED_DAT and LAST_UPDATED_TIM that the Secure Agent uses to run a delta update or delta repeat session. Without delta initialization, the Secure Agent does not have access to the LAST_UPDATED_DAT and LAST_UPDATED_TIM to run a delta update or delta repeat session.

In the INFADI/TBLCHNGN table, how can I view entries corresponding to my mapping run?

You can refer to the session log to find out the session ID for your mapping. In the INFADI/TBLCHNGN table, look for the same session ID to view details about your mapping run. The Secure Agent generates a unique session ID for each mapping run for a particular SAP table. You can also sort the INFADI/TBLCHNGN table entries based on the session ID.

Why does the number of records extracted through the Full or Delta initialization with transfer option not match the number of records extracted through the Delta repeat option?

When you use the **Full** or **Delta initialization with transfer** option, the Secure Agent extracts all the records directly from the SAP table and not only the records that are captured in the change document. However, when you use the **Delta repeat** option, the Secure Agent extracts only the records that are captured in the change document.

Therefore, after you perform a full extraction or delta initialization with transfer, if you run a delta repeat session, the extracted records count might not match with the number of records extracted through the **Full** or **Delta initialization with transfer** option.

Key range partitioning for SAP Table sources

You can configure key range partitioning when you use a mapping task to read data from SAP table sources using normal mode. With key range partitioning, the Secure Agent distributes rows of source data based on the fields that you define as partition keys. The Secure Agent compares the field value to the range values for each partition and sends rows to the appropriate partitions.

Use key range partitioning for columns that have an even distribution of data values. Otherwise, the partitions might have unequal size. For example, a column might have 10 rows between key values 1 and 1000 and the column might have 999 rows between key values 1001 and 2000. If the mapping includes multiple sources, use the same number of key ranges for each source.

When you define key range partitioning for a column, the Secure Agent reads the rows that are within the specified partition range. For example, if you configure two partitions for a column with the ranges as 10 through 20 and 30 through 40, the Secure Agent does not read the rows 20 through 30 because these rows are not within the specified partition range.

You can configure a partition key for fields of the following data types:

- ACCP
- DATS
- INT1
- INT2
- INT4
- NUMC
- TIMS

You cannot use key range partitions when a mapping includes any of the following transformations:

- Web Services
- XML to Relational

Configuring key range partitioning for SAP Table sources

When you use a mapping task to read data from SAP table sources, you can configure key range partitioning to improve performance. Define the partition keys and key ranges based on which the Secure Agent must distribute rows of source data.

1. In the source properties, click the **Partitions** tab.
2. In the **Partition Key** field, select the required partition key from the list.
3. In the **Key Ranges** section, click **Add New Key Range** to define the number of partitions and the key ranges based on which the Secure Agent must partition data.

   Use a blank value for the start range to indicate the minimum value. Use a blank value for the end range to indicate the maximum value.

Best practices for key range partitioning

If you increase the partitions in normal mode, tune the value of the parameter `rdisp/wp_no_btc` on the SAP server accordingly to increase the throughput. The parameter `rdisp/wp_no_btc` denotes the number of background processes.

Contact your SAP administrator to increase the value of the parameter on the SAP side.

SAP Table lookups in mappings

In a mapping, you can configure a Lookup transformation to look up data from SAP Table objects.

If you configure an uncached lookup, you can use only the = logical operator in the lookup condition.

When you configure an uncached lookup, ensure that the data does not contain the pipe (|) character, otherwise data corruption occurs.

When you use an SAP Table object as a lookup, you do not need to configure specific SAP Table properties.

Configuring a mapping with an SAP Table source

Use the Data Integration Mapping Designer to configure a mapping and describe the data flow between the source and target. You can also configure filter and sort options, advanced source properties, key range partitioning, and transformations in the mapping.

1. To create a mapping, click **Data Integration** > **New** > **Mappings**. Select **Mapping** and click **Create**.
2. Enter a name and description for the mapping, and click **OK**.

   You can use alphanumeric characters and underscores (_) in the mapping name.
3. To configure a source, on the **Transformation** palette, click **Source**.
4. In the **Properties** panel, on the **General** tab, enter a name and description.
5. Click the **Source** tab and configure source details.
6. Specify the source type. You can choose one of the following options:

- Select **Single Object** to select a single SAP object.
- Select **Multiple Objects** to specify source object, related source object, and configure the relationship between the source objects. You can use custom relationships to join multiple source objects. When you create a custom relationship for SAP Table objects, you can select the type of join and the source fields to use.
- Select **Parameter** to configure the source objects in a mapping task associated with this mapping.

7. Click **Query Options** in the **Source** tab to specify any filter and sort options for the SAP object.
8. Click **Advanced** to specify the advanced source properties.
9. To add or remove source fields, to update field metadata, or to synchronize fields with the source, click the **Fields** tab.

   Note: You can edit the type, precision, and scale in the SAP table source object metadata.
10. To configure key range partitioning, click the **Partitions** tab.
    a. In the **Partition Key** field, select the required partition key from the list.
    b. In the **Key Ranges** section, click **Add New Key Range** to define the number of partitions and the key ranges based on which the Secure Agent must partition data.

       Use a blank value for the start range to indicate the minimum value. Use a blank value for the end range to indicate the maximum value.
11. To add a transformation, on the **Transformation palette**, click the transformation name. Or, drag the transformation onto the mapping canvas.
    a. On the **General** tab, you can enter a name and description for the transformation.
    b. Draw a link to connect the previous transformation to the transformation.

       When you link transformations, the downstream transformation inherits the incoming fields from the previous transformation.
       For a Joiner transformation, draw a master link and a detail link.
    c. To preview fields, configure the field rules, or rename fields, click **Incoming Fields**.
    d. Configure additional transformation properties, as needed.

       The properties that you configure vary based on the type of transformation you create.
    e. To add another transformation, repeat these steps.
12. To add a Target transformation, on the **Transformation palette**, click **Target**.
    a. On the **General** tab, you can enter a name and description.
    b. Draw a link to connect the previous transformation to the Target transformation.
    c. Click the **Target** tab and configure target details. If necessary, configure the advanced target properties.

       Target details and advanced target properties appear based on the connection type. For more information, see *Transformations*.
    d. To preview fields, configure the field rules, or rename fields, click **Incoming Fields**.
    e. Click **Field Mapping** and map the fields that you want to write to the target.
    f. To add another Target transformation, repeat these steps.
13. Save and run the mapping or save and create a mapping task.

Creating a mapping task

You can create a mapping task based on a valid mapping or integration template on the Mappings page.

1. To create a mapping task, click **Data Integration** > **New** > **Tasks** and then complete one of the following steps:
   - To create a mapping task based on a mapping, select **Mapping Task** and click **Create**.
   - To create a mapping task using a template, expand the appropriate template category and select the template you want to use, and then click **Create**.

   To edit a mapping task, on the **Explore** page, navigate to the mapping task. In the row that contains the task, click **Actions** and select **Edit**.
2. Enter a name for the task.

   Task names must be unique within the organization. Task names can contain alphanumeric characters, spaces, and the following special characters:_ . + -Task names are not case sensitive.
3. Select the runtime environment that contains the Secure Agent that you want to use to access the SAP tables.
4. Select **Mapping** as the task based on which you want to create the mapping task.
5. Click **Select** to specify a mapping.

   The **Select a Mapping** dialog box appears.
6. Select a mapping or search for the required mapping and select **OK**.

   The image of the selected mapping appears.
7. Click **Next**.

   If you specified any parameters in the source or target details in the mapping, the Source or Target page appears. If not, the Schedule page appears.
8. Click **Next** to configure a schedule and advanced options. Perform any of the following steps based on your requirements.
   a. Click **Run this task on schedule** and specify the schedule you want to use.
   b. Configure the email notification options.
   c. Configure advanced options for the task.
   d. Configure the advanced source properties and advanced target properties.
   e. Specify the execution mode.
9. Optionally, add advanced session properties.
   a. Click **Add**.
   b. Select a session property.
   c. Configure the value of the session property.
10. Save and run the mapping task.

Mapping with an SAP Table source example

You can create a mapping to read data from a single SAP object and write the data to a target object.

You can read data from an SAP purchasing document header, the EKKO table, and write the purchasing details to any target.

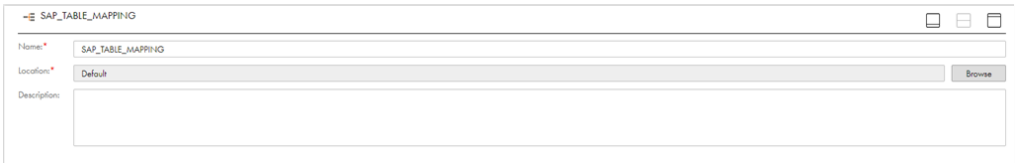
In this example to read data from the EKKO table and write the data to a flat file target object, perform the following steps:

1. Define the mapping.
2. To configure an SAP Table source, select an SAP Table connection and select the EKKO table.
3. To configure a flat file target, select a flat file connection, specify a flat file object, and map the source and target fields.
4. Save the mapping and create a mapping task.

Step 1: Define the mapping

1. To create a mapping, click **Data Integration** > **New** > **Mappings**. Select **Mapping** and click **Create**.
2. Enter a name and description for the mapping.

   The following image shows the **New Mapping** dialog box:



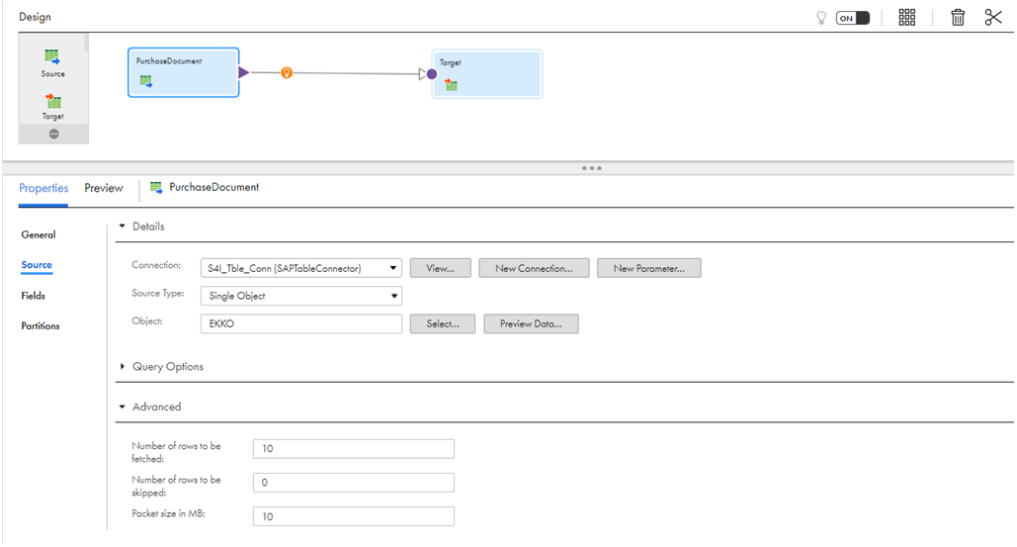
3. Click **OK**.

Step 2: Configure the SAP Table source

1. To configure an SAP source, on the **Transformation** palette, click **Source**.
2. In the **Properties** panel, on the **General** tab, enter a name and description.
3. Click the **Source** tab to configure source details.
4. Specify an SAP Table connection as the source object connection.
5. Specify the source type as **Single Object** and click **Select**.
6. In the **Select Source Object** dialog box, select the EKKO table,
7. Click **Query Options** in the **Source** tab to specify any filter and sort options for the SAP Table object.

8. Click **Advanced** to specify the advanced source properties.

   The following image shows the source details page:

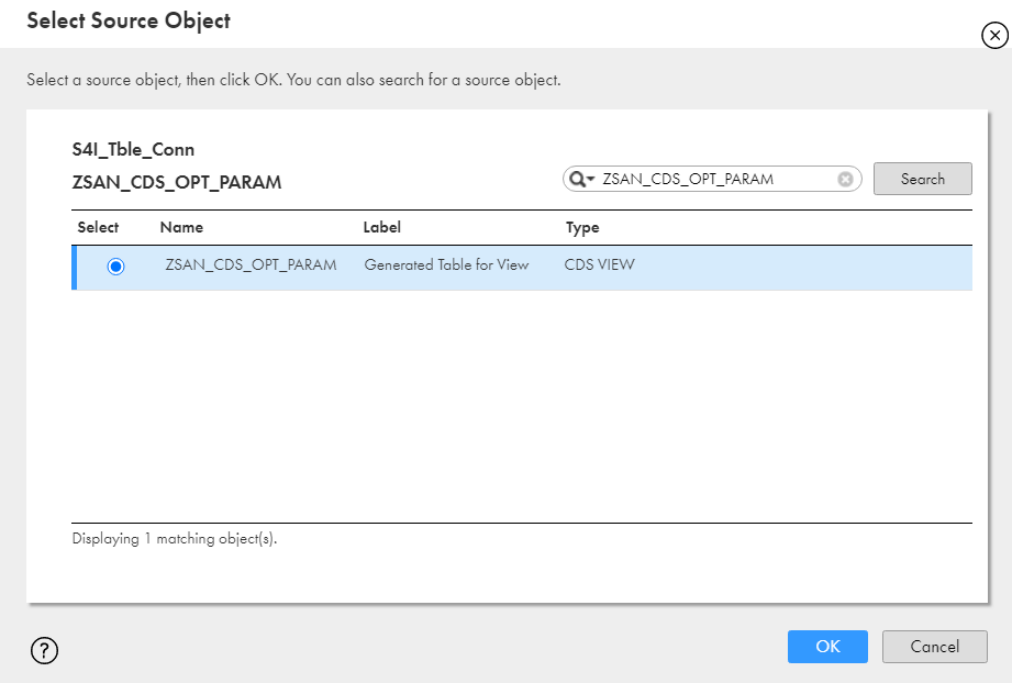


Configure an ABAP CDS view as an SAP source

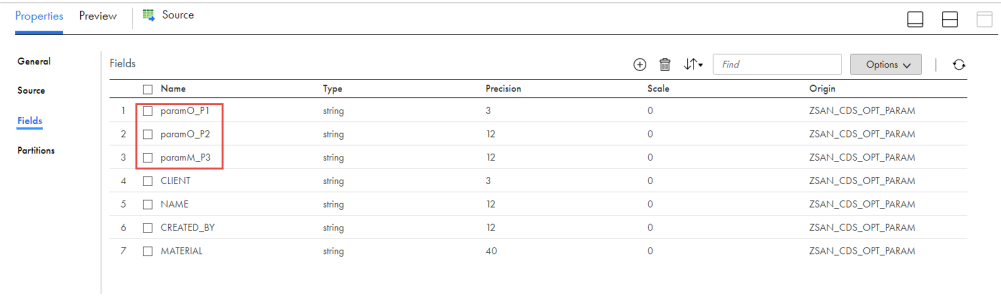
To read from an ABAP CDS view, configure the ABAP CDS view as the source in a mapping.

1. On the **Transformation** palette, click **Source**.
2. In the **Properties** panel, on the **General** tab, enter a name and description.
3. Click the **Source** tab to configure source details.
4. Specify an SAP Table connection as the source object connection.
5. Specify the source type as **Single Object** and click **Select**.
6. In the **Select Source Object** dialog box, select the CDS view object:

The following image displays a CDS view object:

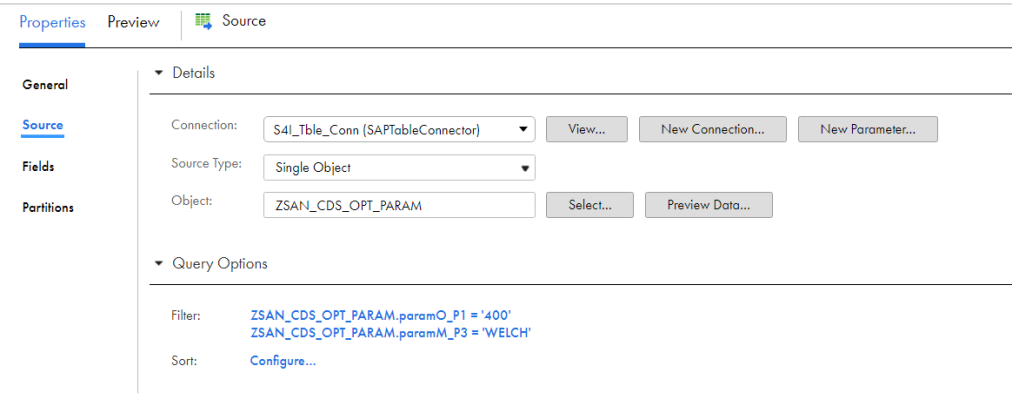


7. When you select a CDS view, on the **Fields** tab, you can view the mandatory and optional parameters:



8. Click **Query Options** in the **Source** tab to specify any filter and sort options for the CDS view:

   The following image shows the basic filter options configured for the CDS view:

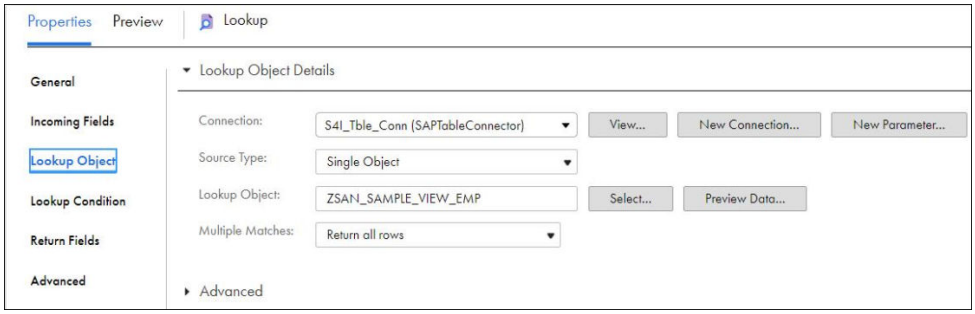


9. Click **Advanced** to specify the advanced source properties.

Configure an ABAP CDS view as a lookup

You can configure an uncached lookup in a mapping to look up data in an ABAP CDS view.

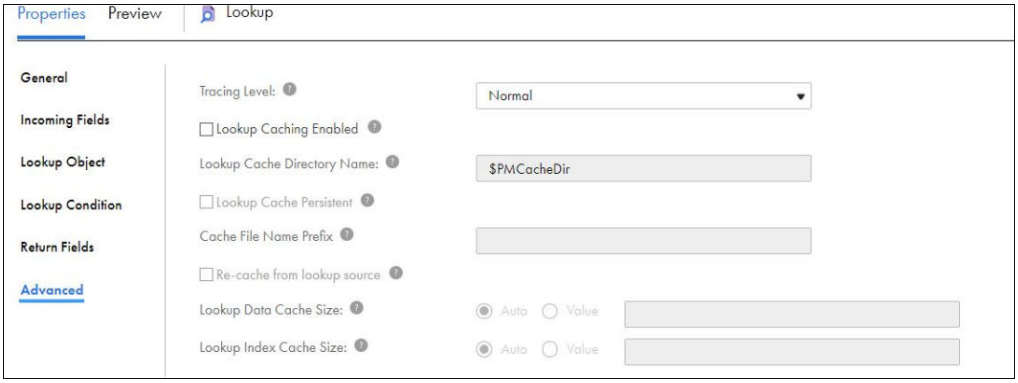
1. From the **Transformation** palette, add a lookup.
2. In the **Properties** panel, on the **General** tab, enter a name and description.
3. On the **Lookup Object** tab, configure the lookup object details.
   a. Specify an SAP Table connection for the lookup object.
   b. Specify the source type as **Single Object**.
   c. In the **Lookup Object** field, click **Select**, and then select the CDS view object as the lookup object.



4. On the **Lookup Condition** tab, specify the lookup condition:



5. On the **Advanced** tab, do not select the **Lookup Caching Enabled** checkbox, and then specify the lookup properties:

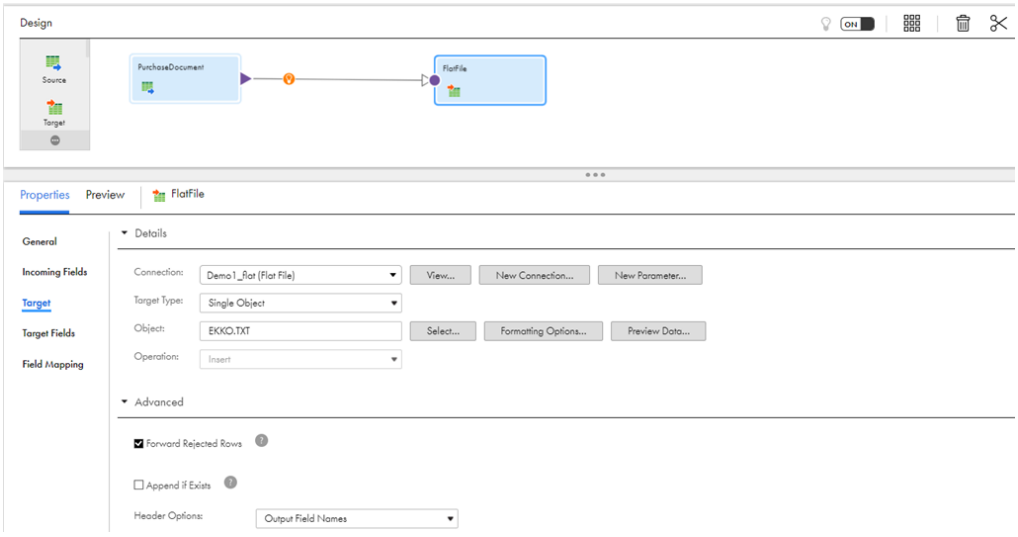


Step 3: Configure the flat file target

1. To add a flat file Target transformation, on the **Transformation** palette, click **Target**.
2. On the **General** tab, enter a name and description.

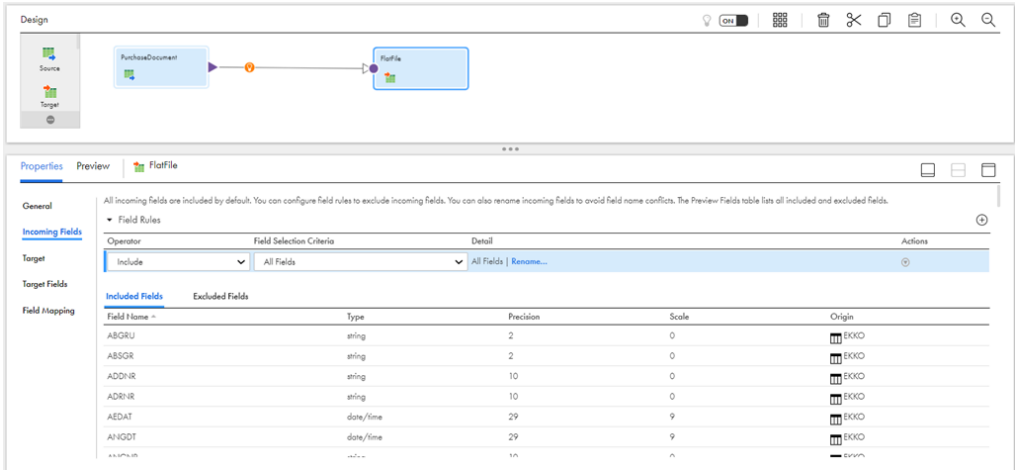
3. Draw a link to connect the Source transformation to the Target transformation.
4. Click the **Target** tab to configure the flat file target details.
5. Specify a flat file connection as the target connection.
6. Select the target type as **Single Object** and click **Select**.
7. Specify a flat file object.

   The following image shows the target details:



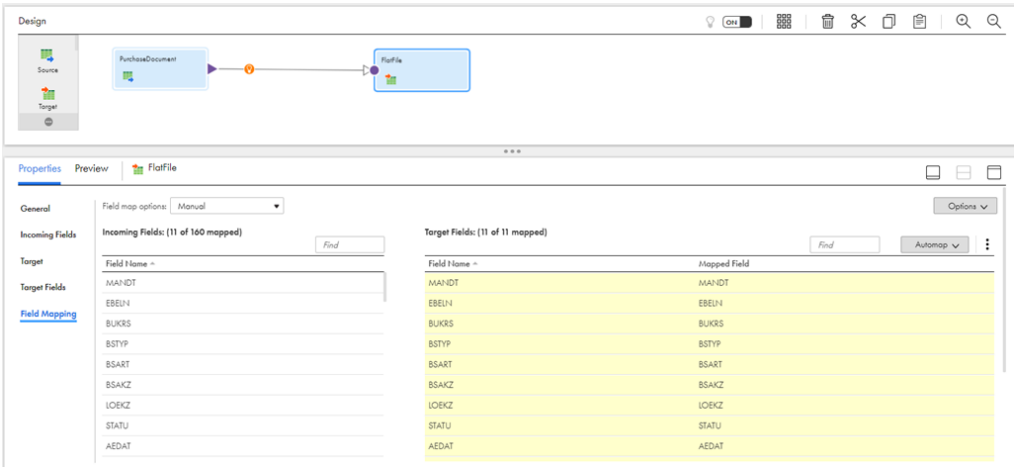
8. To preview fields, click **Incoming Fields**.

   The following image shows the incoming field details:



9. Click **Field Mapping** and map the fields that you want to write to the target.

   The following image shows the field mapping details:

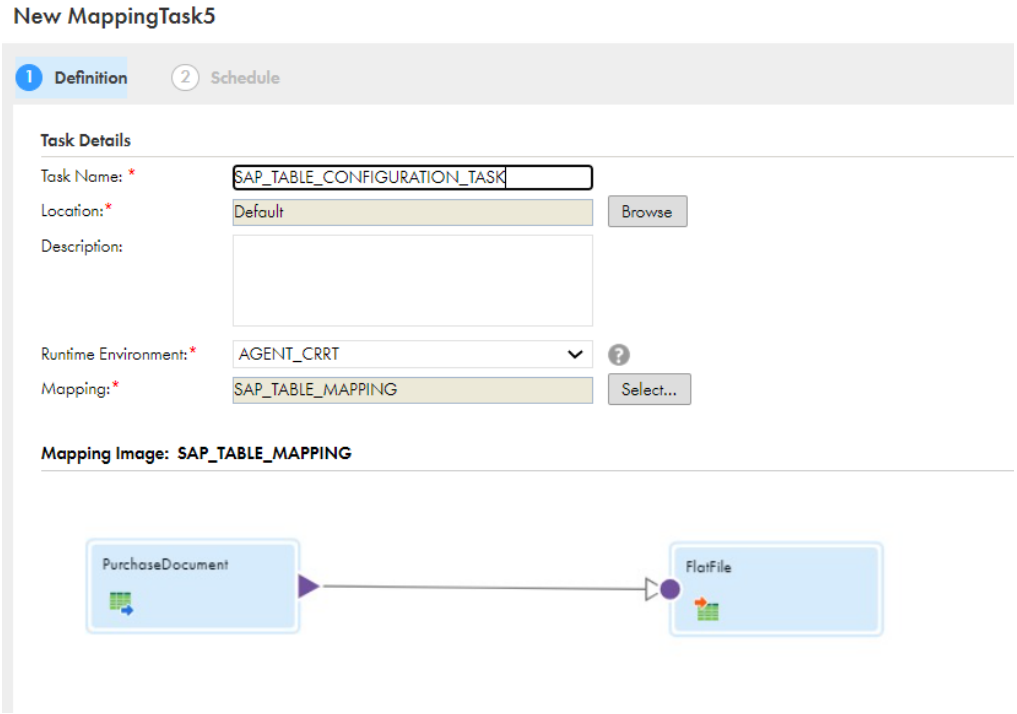


Step 4: Save the mapping and create a mapping task

1. Click **Save** > **Save and New Configuration Task**.

   The **New Mapping Task** page appears.
2. Enter a name and description for the task.
3. Select the runtime environment that contains the Secure Agent you want to use to access SAP tables.

   The following image shows the mapping task details:



4. Click **Next** to configure the schedule and advanced options.
5. Save and run the mapping task.

CHAPTER 4

Synchronization tasks with SAP Table

The Synchronization application allows you to synchronize data between a source and target.

You can configure a synchronization task using the Synchronization Task wizard. You can use SAP Table objects as sources, targets, or lookup objects. You can use expressions to transform the data according to your business logic, use data filters to filter data before writing it to targets, sort data in ascending or descending order of multiple fields.

When you create a task, you can associate it with a schedule to run it at specified times or on regular intervals. Or, you can run it manually. You can monitor tasks that are currently running and view logs about completed tasks.

SAP Table sources in synchronization tasks

When you configure a synchronization task to use an SAP Table source, you can configure the source properties.

The source properties appear on the **Source** page of the Synchronization Task wizard when you specify an SAP Table connection.

The following table describes the SAP Table source properties:

Property	Description
Connection	Name of the source connection.
Source Type	Source type. Select one of the following types: - Single. Select to specify a single SAP Table object. - Multiple. Select to specify multiple SAP Table objects. When you specify multiple source objects, you need to create relationships between the source objects.
Source Object	Source object for the task.
Add	Adds multiple source objects.
Create Relationship	Creates relationship between selected source object and related source object. Specify a join condition between a source object key field and a related source object key field.

Property	Description
Edit Relationship	Edits a join condition.
Display technical field names instead of labels	When selected, displays technical names instead of business names of the fields in the specified source object.
Display source fields in alphabetical order	When selected, displays source fields in alphabetic order. By default, fields appear in the order returned by the source system.
Data Preview	Displays the first 10 rows of the first five columns in the object and the total number of columns in the object.
Preview All Columns	Previews all source columns in a file.

You can also configure advanced source properties when you schedule the synchronization task. Advanced source properties appear on the **Schedule** page of the Synchronization Task wizard.

The following table describes the SAP Table advanced source properties:

Property	Description
Number of rows to be fetched	The number of rows that are randomly retrieved from the SAP Table. Default value of zero retrieves all the rows in the table.
Number of rows to be skipped	The number of rows to be skipped.
Packet size in MB	Packet size. Default is 10 MB.
Enable Compression	Enables compression. If the Secure Agent and the SAP System are not located in the same network, you may want to enable the compression option to optimize performance.

SAP Table lookups in synchronization tasks

When you configure field mappings in a synchronization task, you can create a lookup to an SAP Table object.

If you configure an uncached lookup, you can use only the = logical operator in the lookup condition.

When you configure an uncached lookup, ensure that the data does not contain the pipe (|) character, otherwise data corruption occurs.

When you use an SAP Table object as a lookup, you do not need to configure specific SAP Table properties.

Configuring a synchronization task with a single SAP object as the source

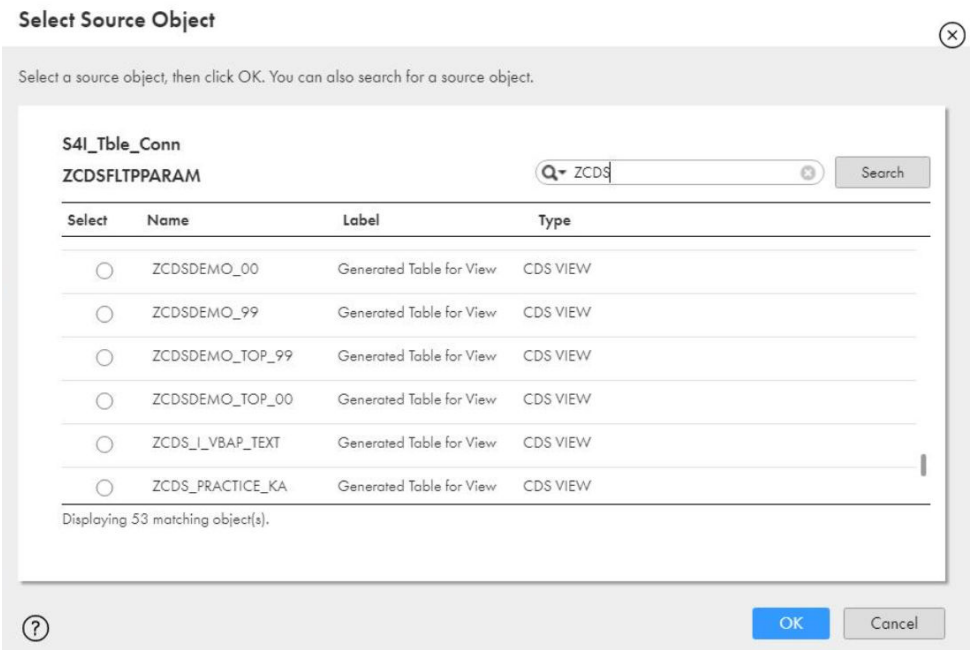
1. To create a synchronization task, click **Data Integration** > **New** > **Tasks**. Select **Synchronization Task** and click **Create**.
2. Enter a name for the synchronization task.

   The names of synchronization tasks must be unique within the organization. Synchronization task names can contain alphanumeric characters, spaces, and the following special characters:_ . + -

   Synchronization task names are not case sensitive.
3. Enter a description for the synchronization task.

   The description can have a maximum length of 255 characters.
4. Select the task operation that you can perform on the target. Select one of the following options: Insert, Update, Upsert, and Delete.
5. Click **Next** to enter the source details.
   a. Select an SAP Table connection.
   b. Select **Single** as the source type.
   c. Click **Select** to specify the SAP source object.

      The **Select Source Object** dialog box appears. The dialog box displays up to 200 objects. If the objects you want to use do not appear, enter a search string to search based on name and description.
      The following image displays the CDS views that you can select:



   d. Click **Select**.

      The Data Preview area displays the first 10 rows of the first five columns in the SAP object and the total number of columns in the object. To preview all source columns in a file, click **Preview All Columns**.

6. To display technical names instead of business names, select **Display technical field names instead of labels**.
7. To display source fields in alphabetic order, click **Display source fields in alphabetical order**.

   By default, fields appear in the order returned by the source system.
8. Click **Next** to specify the target connection and target objects.
9. Click **Next** to specify any data filters or sort criteria.

   Note: Specify the row limit in the **Advanced Source Properties** section in the **Schedule** page.
10. Click **New** to create a data filter. You can choose to create a simple or advanced data filter.
    - To create a simple data filter, select a source object, source field, and operator. Enter the value you want to use and click **OK**.
    - To create an advanced data filter, click **Advanced**. Select a source object and enter the field expression you want to use and click **OK**.
      You can use parameters defined in a parameter file in the data filters. When you use a parameter in a data filter, start the data filter with the parameter.
11. Click **New** to configure the sort criteria.
    a. Select the source object, sort by field, and the sort direction.
    b. Click **New** to configure additional sort criteria or click **Delete** to remove a sort criteria.
12. Click **Next** to configure the field mappings. Perform any of the following steps based on your requirements.
    a. Click **Edit Types** in the Source column to edit the precision and scale of the SAP object.
    b. Click **Add Mapplet** to select a mapplet and optionally specify a connection for the mapplet.
    c. Click **Automatch** to match source and target fields with similar names.
    d. Click **Refresh Fields** to update the cache and view the latest field attributes.
    e. Click **Edit Types** in the Target column to edit the data type, precision, and scale of the target object. Note that this option is not available for all target types.
    f. Select a source field and drag it to the target field to map the source and target fields. Repeat for all the fields that you want to map.
    g. Click the **Add or Edit Expression** icon to define a field expression to transform data.
    h. Click the **Add or Edit Lookup** icon to create a lookup. Specify the lookup connection, object, source and lookup fields, output field, multiplicity, and lookup expression.
    i. Click **Validate Mapping** to validate all the field mappings.
    j. Click **Clear Mapping** to clear all the field mappings.
13. Click **Next** to configure a schedule and advanced options. Perform any of the following steps based on your requirements.
    a. Click **Run this task on schedule** and specify the schedule you want to use.
    b. Configure the email notification options.
    c. Configure advanced options for the task.
    d. Configure the advanced source properties and advanced target properties.
    e. Specify the execution mode.
14. Save the synchronization task. You can choose one of the following save options.
    - Click **Save and Close** to save the task and close the synchronization task.
    - Click **Save and Continue** to save the task and continue with configuring the synchronization task.

- Click **Save and Run** to save and run the synchronization task.

Configuring a synchronization task with multiple SAP objects as the source

1. To create a synchronization task, click **Data Integration** > **New** > **Tasks**. Select **Synchronization Task** and click **Create**.
2. Enter a name for the synchronization task.

   The names of synchronization tasks must be unique within the organization. Synchronization task names can contain alphanumeric characters, spaces, and the following special characters:_ . + -

   Synchronization task names are not case sensitive.
3. Enter a description for the synchronization task.

   The description can have a maximum length of 255 characters.
4. Select the task operation that you can perform on the target. Select one of the following options: Insert, Update, Upsert, and Delete.
5. Click **Next** to enter the source details.
   a. Select an SAP Table connection.
   b. Select **Multiple** as the source type.
   c. Click **Add** to specify an SAP source object.

      The **Select Source Object** dialog box appears. The dialog box displays up to 200 objects. If the objects you want to use do not appear, enter a search string to search based on name and description. To search for an object using the technical name, enclose the name in double quotes.
   d. Repeat the previous steps to add multiple SAP objects. To remove a selected object, click the **Delete** icon.
6. Create relationships between the multiple SAP objects.
   a. Select an SAP object and click **Create Relationship** to create the join conditions between the source and the related object.

      The **Create Relationship** dialog box appears.
   b. Specify the key field in the source SAP object, the type of join, the join operator, the related SAP object, and the key field in the related object.
   c. Click **OK** to create the relationship.
   d. Repeat the previous steps to create multiple relationships.
7. To display technical names instead of business names, select **Display technical field names instead of labels**.
8. To display source fields in alphabetic order, click **Display source fields in alphabetical order**.

   By default, fields appear in the order returned by the source system.
9. Click **Next** to specify the target connection and target objects.
10. Click **Next** to specify any data filters or sort criteria.

    Note: Specify the row limit in the **Advanced Source Properties** section in the **Schedule** page.
11. Click **New** to create a data filter. You can choose to create a simple or advanced data filter.

- To create a simple data filter, select a source object, source field, and operator. Enter the value you want to use and click **OK**.
- To create an advanced data filter, click **Advanced**. Select a source object and enter the field expression you want to use and click **OK**.
  You can use parameters defined in a parameter file in data filters. When you use a parameter in a data filter, start the data filter with the parameter.

12. Click **New** to configure the sort criteria.
    a. Select the source object, sort by field, and the sort direction.
    b. Click **New** to configure additional sort criteria or click **Delete** to remove a sort criteria.
13. Click **Next** to configure the field mappings. Perform any of the following steps based on your requirements.
    a. In the Source column, select one of the SAP objects or All source objects to map the fields.
    b. Click **Edit Types** in the Source column to edit the precision and scale of the selected SAP object.
    c. Click **Add Mapplet** to select a mapplet and optionally specify a connection for the mapplet.
    d. Click **Automatch** to match source and target fields with similar names.
    e. Click **Refresh Fields** to update the cache and view the latest field attributes.
    f. Click **Edit Types** in the Target column to edit the data type, precision and scale of the target object. Note that this option is not available for all target types.
    g. Select a source field and drag it to the target field to map the field. Repeat for all the fields that you want to map.
    h. Click the **Add or Edit Expression** icon to define a field expression to transform data.
    i. Click the **Add or Edit Lookup** icon to create a lookup. Specify the lookup connection, object, source and lookup fields, output field, multiplicity, and lookup expression.
    j. Click **Validate Mapping** to validate all the field mappings.
    k. Click **Clear Mapping** to clear all the field mappings.
14. Click **Next** to configure a schedule and advanced options. Perform any of the following steps based on your requirements.
    a. Click **Run this task on schedule** and specify the schedule you want to use.
    b. Configure the email notification options.
    c. Configure advanced options for the task.
    d. Configure the advanced source properties and advanced target properties.
    e. Specify the execution mode.
15. Save the synchronization task. You can choose one of the following save options.
    - Click **Save and Close** to save the task and close the synchronization task.
    - Click **Save and Continue** to save the task and continue with configuring the synchronization task.
    - Click **Save and Run** to save and run the synchronization task.

Monitoring a synchronization task

After you run a synchronization task, you can monitor the task and view the logs.

In **Monitor**, you can monitor the status of the logs after you run the task.

You can also monitor the progress of the task by calling Transaction SM37 from SAP. You can view the actual job duration in SAP. The job duration listed in the Data Integration activity log is a higher value because it also includes time required to complete processing in Data Integration.

You can view the HTTP and HTTPS log files in the SMICM transaction. Optionally, you can increase trace level to 3 to view the detailed logs.

Synchronization task example

You can create a synchronization task to read data from multiple SAP objects and write the data to a flat file object.

You can read General Ledger Accounting line items from the BKPF and BSEG tables in SAP. BSEG is an SAP Cluster table that is used to store Accounting Document Segment information. BKPF is a Transparent SAP Table that is used to store Accounting Document Header information. In this example, you can join the BKPF and BSEG tables and map the source object to a flat file target object.

In this example to write the accounting document details to a flat file object, perform the following steps:

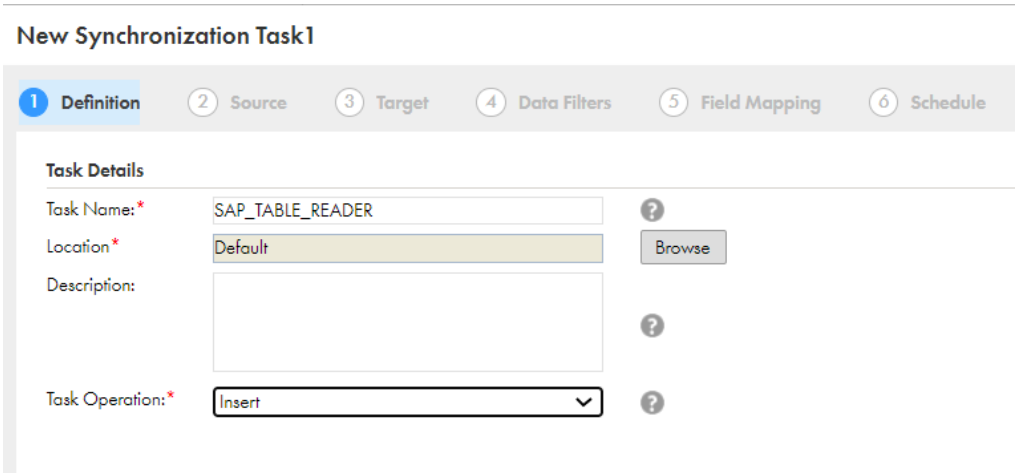
1. Define the synchronization task.
2. To configure the SAP Table sources, select an SAP Table connection, and select the BKPF transparent table and the BSEG cluster table as the source objects. Create join conditions between the source BKPF table and the related BSEG table.
3. To configure a flat file target for the task, select a flat file connection and specify a flat file object.
4. Configure the field mappings to define the data that the synchronization task writes to the target.
5. Save and run the synchronization task.

Step 1: Define the synchronization task

1. To create a synchronization task, click **Data Integration** > **New** > **Tasks**. Select **Synchronization Task** and click **Create**.
2. Enter a name for the synchronization task.
3. Enter a description for the synchronization task.

4. Select the insert task operation for the target.

   The following image shows the task definition page:



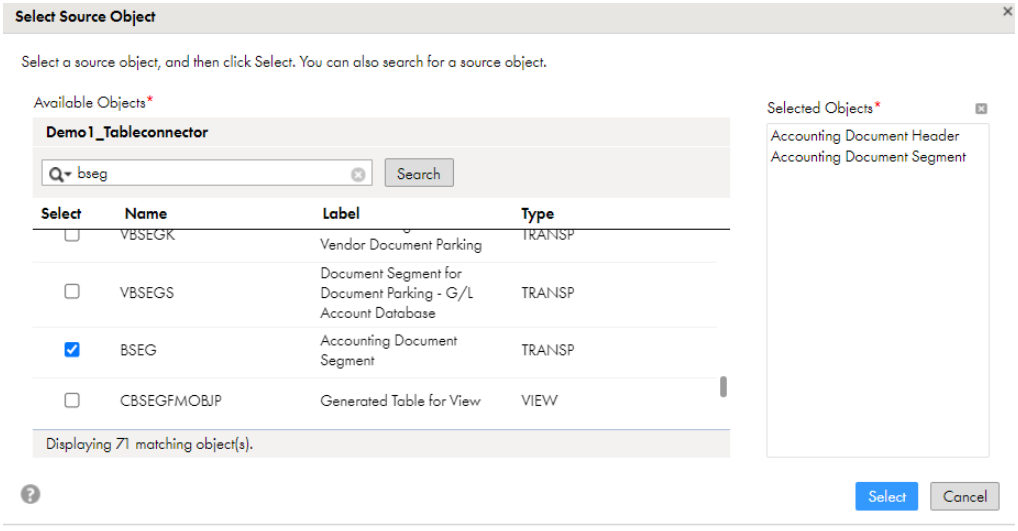
5. Click **Next**.

Step 2: Configure the SAP Table source

1. Select an SAP Table connection.
2. Select **Multiple** as the source type.
3. Click **Add** to specify the SAP source object.

   The **Select Source Object** dialog box appears. Select the BKPF transparent table.
4. Click **Select**.
5. Click **Add** to select the BSEG cluster table.

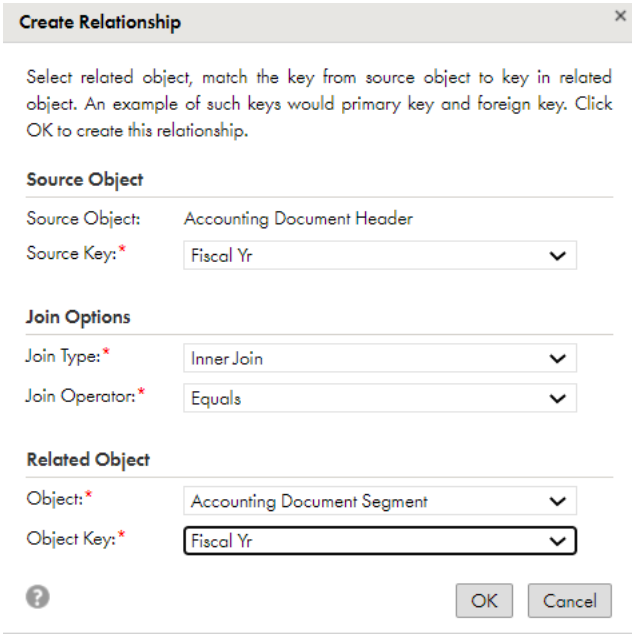
   The following image shows the **Select Source Object** dialog box:



6. Create relationships between the SAP tables.
   a. Select the BKPF SAP object and click **Create Relationship** to create the join conditions between the source BKPF table and the related BSEG table.

      The **Create Relationship** dialog box appears.
   b. Specify the key field in the source SAP object, the type of join, the join operator, the related SAP object, and the key field in the related object.
   c. Click **OK** to create the relationship.
   d. Repeat the previous steps to create multiple relationships.

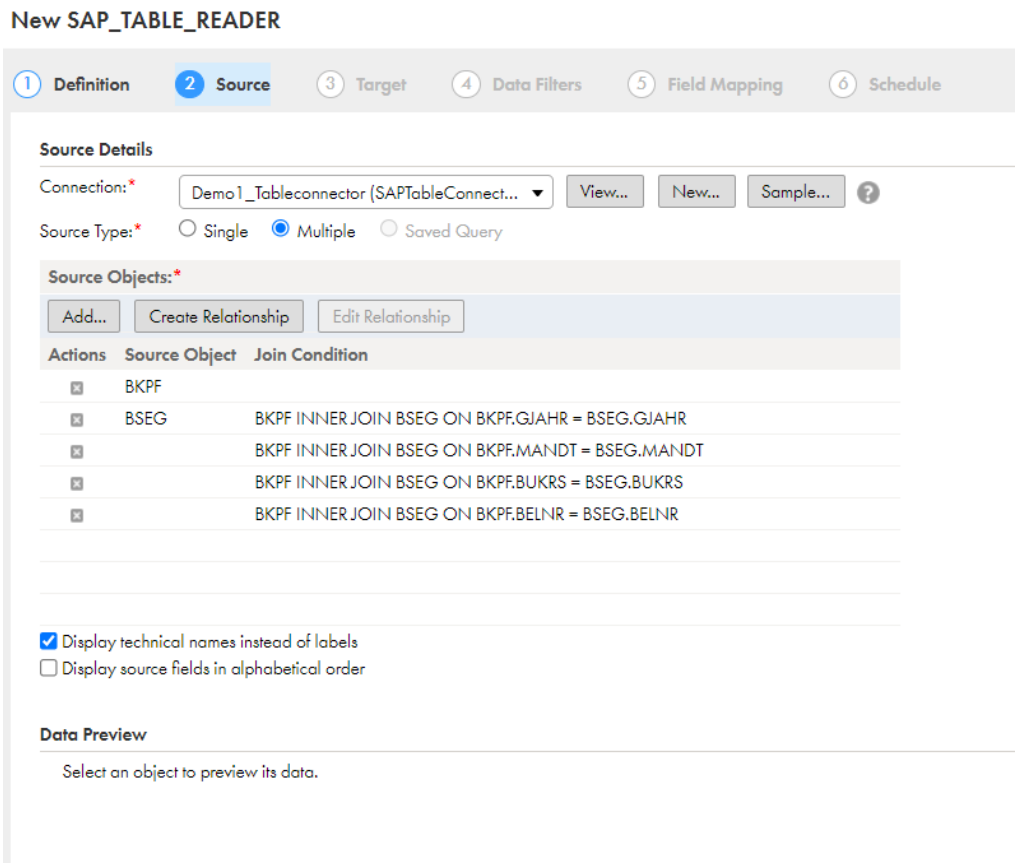
   The following image shows the **Create Relationship** dialog box:



7. Select a source object to preview the data. The Data Preview area displays the first 10 rows of the first five columns in the SAP object. You can also view the total number of columns in the object. To preview all source columns in a file, click **Preview All Columns**.
8. To display technical names instead of business names, select **Display technical field names instead of labels**.
9. To display source fields in alphabetic order, click **Display source fields in alphabetical order**.

By default, fields appear in the order returned by the source system.

The following image shows the join conditions for multiple SAP objects in the task source details page:

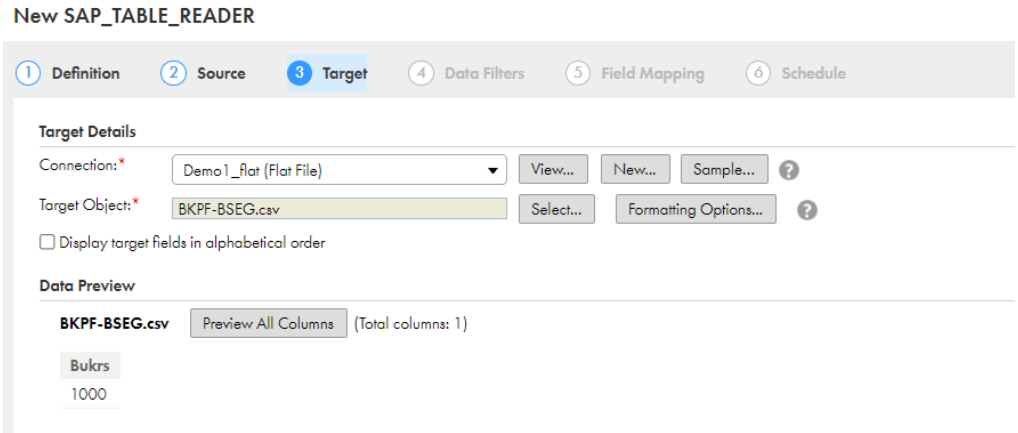


10. Click **Next**.

Step 3: Configure the flat file target

1. Select a flat file connection and select a flat file object.
2. Select a target flat file object and click **OK**.

   The following image shows a flat file object in the task target details page:



3. Click **Next** to specify any data filters or sort fields.
4. Click **Next**.

Step 4: Configure the field mapping

1. Map the source and target fields.

   You can select all source objects or one of the source objects to map with the target fields.
2. Click **Next** to configure a schedule and advanced options.
3. Save and run the synchronization task.

CHAPTER 5

Data type reference

Data Integration uses the following data types in mappings, synchronization tasks, and mapping tasks with SAP:

Native data types

Native data types are data types specific to the source and target databases or flat files. They appear in non-SAP sources and targets in the mapping.

SAP data types

SAP data types appear in the Fields tab for Source and Target transformations when you choose to edit metadata for the fields. SAP performs any necessary conversion between the SAP data types and the native data types of the underlying source database tables.

Transformation data types

Set of data types that appear in the remaining transformations. They are internal data types based on ANSI SQL-92 generic data types, which Data Integration uses to move data across platforms. Transformation data types appear in all remaining transformations in a mapping, synchronization task, or mapping task.

When Data Integration reads source data, it converts the native data types to the comparable transformation data types before transforming the data. When Data Integration writes to a target, it converts the transformation data types to the comparable native data types.

SAP and transformation data types

The following table lists the SAP data types that SAP Table Connector supports along with the corresponding transformation data types:

SAP Data Type	Transformation Data Type	Range for Transformation Data Type
ACCP	Date/time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D.
CHAR	String	1 to 104,857,600 characters Fixed-length or varying-length string.
CLNT	String	1 to 104,857,600 characters Fixed-length or varying-length string.

SAP Data Type	Transformation Data Type	Range for Transformation Data Type
CUKY	String	1 to 104,857,600 characters Fixed-length or varying-length string.
CURR	Decimal	Precision 1 to 28 digits, scale 0 to 28
DATS	Date/time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. Precision to the nanosecond.
DEC	Decimal	Precision 1 to 28 digits, scale 0 to 28
DF16_DEC	Decfloat16	Range of 1-15 and scaling of maximum 14. Decimal floating point number stored in BCD format. You can use the DF16_DEC data type when you read data from SAP tables.
DF34_DEC	Decfloat34	Range of 1-31 and scaling of maximum 30. Decimal floating point number stored in BCD format. You can use the DF34_DEC data type when you read data from SAP tables.
DF16_RAW	Double	Maximum of 16 positions with floating decimal. Decimal floating point number stored in binary format. You can use the DF16_RAW data type when you read data from SAP tables.
DF34_RAW	Double	Maximum of 34 positions with floating decimal. Decimal floating point number stored in binary format. You can use the DF34_RAW data type when you read data from SAP tables.
FLTP	Double	Precision 15, scale 0
INT1	Small Integer	Precision 5, scale 0
INT2	Small Integer	Precision 5, scale 0
INT4	Integer	Precision 10, scale 0
INT8	Int8	8-byte integer between -9,223,372,036,854,775,808 and +9,223,372,036,854,775,807. The length is set at 19 positions. Use the INT8 data type when you read data from and write data to SAP tables.
LANG	String	1 to 104,857,600 characters Fixed-length or varying-length string.
LCHR	String	1 to 104,857,600 characters Fixed-length or varying-length string.
LRAW	Binary	Uninterrupted sequence of bytes with a maximum length of 255 positions.
NUMC	String	1 to 104,857,600 characters Fixed-length or varying-length string.
PREC	Binary	Uninterrupted sequence of bytes with a maximum length of 255 positions.
QUAN	Decimal	Precision 1 to 28 digits, scale 0 to 28

SAP Data Type	Transformation Data Type	Range for Transformation Data Type
RAW	Binary	Uninterrupted sequence of bytes with a maximum length of 255 positions.
RAWSTRING	Binary	Uninterrupted byte string. You can use the RAWSTRING data type when you read data from SAP tables.
SSTRING	String	Small Character string. You can use the SSTRING data type when you read data from SAP tables.
STRING	String	Character string. You can use the STRING data type when you read data from SAP tables.
TIMS	Date/time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. Precision to the nanosecond.
UNIT	String	1 to 104,857,600 characters Fixed-length or varying-length string.
VARC	String	1 to 104,857,600 characters Fixed-length or varying-length string.

Rules and guidelines for SSTRING, STRING, and RAWSTRING data types

If you import metadata that contains a SSTRING, STRING or RAWSTRING data type with precision that is not defined in SAP, you need to set the **SapTableReaderPrecision** custom property for the Secure Agent and specify the required precision.

Perform the following steps to configure the custom property for the Secure Agent:

1. Log on to Informatica Intelligent Cloud Services.
2. Click **Administrator**.
3. In the navigation bar, select the **Runtime Environments** tab.
4. In the Runtime Environments page, select the Secure Agent used for running the SAP task.
5. Select **Edit** on the top right-hand corner.
6. In the **System Configuration Details** section, add a custom property.
7. Select **Service** as **Data Integration Service**, and then select **Type** as **Tomcat**.
8. In the **Name** field, specify **SapTableReaderPrecision**, and in the **Value** field, set the required precision.
9. Click **Save**.

You can't use the SSTRING, STRING or RAWSTRING data type in a task when you write to SAP table at this time. Tasks that include these data types for the SAP table writer might fail.

CHAPTER 6

FAQ for SAP Table Connector

How can I avoid the CPIC error in the SAP instance when I run a mapping?

For more information about the issue and steps to avoid the CPIC error, see the following Informatica Knowledge Base article: KB 000176711.

How to solve the following error that occurs when I import metadata for an SAP table at design time:
`"OPTION_NOT_VALID: OPTION_NOT_VALID Message 000 of class SAIS type E"`

For more information about the issue when you import metadata for an SAP table at design time, see the following Informatica Knowledge Base article: KB 000174054.

How can I avoid truncation of the tab character when the CHAR data type in an SAP table contains a tab character at the end?

For more information about the issue and steps to avoid the truncation error, see the following Informatica Knowledge Base article: KB 000179163.

How can I improve performance of an SAP Table mapping that is configured for delta extraction for a large volume of data?

For more information about improving performance of a mapping configured for delta extraction, see the following Informatica Knowledge Base article: KB 000184837.

How to solve the following error that occurs when I process the query to look up or filter data from the SAP Table object: `An exception with the type CX_SY_DYNAMIC_OSQL_SYNTAX occurred, but was neither handled locally, nor declared in a RAISING clause.`

For more information about the issue and steps to avoid the error, see the following Informatica Knowledge Base article: KB 000185754.

How can I import the SAP metadata that contains a SSTRING, STRING, or RAWSTRING data type with precision that is not defined in the SAP system?

For more information about importing the SAP metadata that contains a SSTRING, STRING, or RAWSTRING data type with precision that is not defined in the SAP system, see "Rules and guidelines for SSTRING, STRING, and RAWSTRING data types" on page 66.

How to solve the following error that occurs when I use an SAP table as a source object in a synchronization task:
`Field QUERYRESULT not a member of TABLES`

You need to install the latest transport files and clear the browser cache.

INDEX

A

ADSO source objects
  ADSO 11

C

Cloud Application Integration community
  URL 6
Cloud Developer community
  URL 6
configuring key range
  partitioning for SAP table sources 44
connections
  SAP Table 17

D

Data Integration community
  URL 6
data types
  SAP 64

I

Informatica Global Customer Support
  contact information 7
Informatica Intelligent Cloud Services
  web site 6

M

maintenance outages 7
mapping example
  SAP Table source 47
Mapping tasks
  creating 46
  overview 29
mappings
  overview 29
  SAP Table lookups 44
  SAP Table source example 47
  SAP Table sources 29

P

Partitioning
  key range partitioning for SAP table sources 43

S

SAP
  data types 64
SAP integration methods
  using SAP Table 8
SAP sources
  tables and views 8
SAP sources and targets
  rules and guidelines in Synchronization tasks 9
SAP Table
  connection properties 17
SAP Table Connector
  assets 12
  lookup 12
  source 12
  target 12
  transformations 12
SAP Table mapping example
  defining the mapping 47
SAP Table mappings
  configuring 44
status
  Informatica Intelligent Cloud Services 7
Synchronization tasks
  example 59
  monitoring 58
  multiple SAP object sources 57
  overview 53
  rules and guidelines for SAP sources and targets 9
  SAP Table lookups 54
  SAP Table sources 53
  single SAP object source 55
system status 7

T

troubleshooting
  SAP Table 67
trust site
  description 7

U

upgrade notifications 7

W

web site 6