Informatica®

Informatica® Cloud Data Integration

SAP Connector

Informatica Cloud Data Integration SAP Connector
July 2024

© Copyright Informatica LLC 2015, 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 5

Informatica Resources 5

- Informatica Documentation 5
- Informatica Intelligent Cloud Services web site 5
- Informatica Intelligent Cloud Services Communities 5
- Informatica Intelligent Cloud Services Marketplace 5
- Data Integration connector documentation 6
- Informatica Knowledge Base 6
- Informatica Intelligent Cloud Services Trust Center 6
- Informatica Global Customer Support 6

Chapter 1: Introduction to SAP Connector 7

Data integration and SAP integration methods 7

Communication interfaces 8

SAP Connector assets 8

Chapter 2: Connections for SAP 9

Prerequisites 9

- Download and configure the SAP libraries 9
- Configure SAP user authorization 11
- Configure the sapnwrfc.ini file 11
- Define SAP Connector as a logical system in SAP 14

Connect to SAP 17

- Before you begin 17
- Connection details 17
- SAP connection types 17

Use the serverless runtime environment 19

Chapter 3: Data integration using BAPI/RFC functions 21

BAPI/RFC mapplets 21

- BAPI/RFC mapplet parameters 22
- Target object for BAPI/RFC error output 25
- Rules and guidelines for BAPI/RFC mapplets in mappings 25

Mapping and mapping tasks using BAPI/RFC functions 25

- Generating a BAPI/RFC mapplet 26
- Configuring a mapping with a BAPI/RFC mapplet 27
- Mappings with BAPI/RFC function example 28
- SAP mapplet properties 32

Chapter 4: Data integration using IDocs. 34
IDoc mapplets. 35
Segments and groups. 35
Outbound mapplet. 38
Inbound mapplet. 39
Configuring IDoc mapplets. 41
Mapping and mapping tasks using IDocs. 42
IDoc reader sources in mappings. 42
IDoc writer targets in mappings. 44
Configuring IDoc mapplets. 44
Configuring an outbound mapping to read IDocs from SAP. 45
Configuring an inbound mapping to write IDocs to SAP. 46
Outbound mapping to read MATMAS IDocs from SAP example. 47
Inbound mapping to write IDocs to SAP example. 50
SAP mapplet properties. 53
Chapter 5: SAP data type reference. 55
SAP and transformation data types. 55
Index. 58

Preface

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

You can use SAP Connector to integrate with SAP systems in batch, asynchronous, or synchronous modes based on your requirements.

Use SAP Connector to read data from or write data to SAP through the Intermediate Documents (IDocs) or BAPI/RFC interface. You can choose an SAP connection type to connect to SAP systems based on the interface requirements.

SAP integrates multiple business applications and solutions, such as Customer Relationship Management (CRM), Advanced Planner and Optimizer (APO), and Bank Analyzer. Developers can add business logic within SAP using Java 2 Enterprise Edition (J2EE) or Advanced Business Application Programming-Fourth Generation (ABAP/4 or ABAP), a language proprietary to SAP.

You can use the SAP connection in mappings and mapping tasks. Create a mapping task to process data based on the data flow logic defined in a mapping or integration template. You can switch mappings to advanced mode to include transformations and functions that enable advanced functionality.

Data integration and SAP integration methods

You can use different integration methods to read from and write to SAP at the different level of integrations.

The following table lists the integration method corresponding to the different levels and connections:

SAP Integration Method	Level of Integration	Connections
SAP ADSO	Object-level integration	SAP Table and SAP ADSO Writer connections using a Secure Agent or serverless runtime environment. The SAP Table connection information is available in the SAP Table Connector documentation. For more information about the SAP Table connection, see the *SAP Table Connector* documentation. The SAP ADSO Writer connection information is available in the SAP ADSO Writer Connector documentation. For more information about the SAP ADSO Writer connection, see the *SAP ADSO Writer Connector* documentation.
SAP BAPI functions	Object-level integration	SAP BAPI connection using a Secure Agent or serverless runtime environment. The SAP BAPI connection information is available in the Data Integration Connections documentation. For more information, see *SAP BAPI connection* in the Data Integration Connections help.

SAP Integration Method	Level of Integration	Connections
SAP BAPI/RFC functions	Object-level integration	SAP RFC/BAPI Interface connection using a Secure Agent or serverless runtime environment.
SAP BW	Object-level integration	SAP BW connection using a Secure Agent. The SAP BW connection information is available in the SAP BW Connector documentation. For more information about the SAP BW connection, see the *SAP BW Connector* documentation.
SAP IDoc	Message-level integration	IDoc Reader and IDoc Writer connections using a Secure Agent or serverless runtime environment.
SAP ODP	Object-level integration	SAP ODP Extractor connection using a Secure Agent or serverless runtime environment. The SAP ODP Extractor connection information is available in the SAP ODP Extractor Connector documentation. For more information about the SAP ODP Extractor connection, see the *SAP ODP Extractor Connector* documentation.
SAP Table	Data-level integration	SAP Table connection using a Secure Agent or serverless runtime environment. The SAP Table connection information is available in the SAP Table Connector documentation. For more information about the SAP Table connection, see the *SAP Table Connector* documentation.

Communication interfaces

SAP uses TCP/IP as the native communication interface to communicate with Data Integration.

SAP also uses the Remote Function Call (RFC) communication protocol to communicate with Data Integration. To execute remote calls from Data Integration, SAP requires connection information, and the service name and gateway on the application server. The service and gateway parameters, and connection information is stored in a configuration file named `sapnwrfc.ini` on the Secure Agent machine.

SAP Connector assets

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

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

- Mapping
- Mapping task

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

CHAPTER 2

Connections for SAP

Create an SAP connection to access data from SAP through the Intermediate Documents (IDocs) or BAPI/RFC interface.

You can use an SAP connection in mappings and mapping tasks.

Prerequisites

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

To process IDocs and SAP BAPI/RFC functions, you also need to verify if the required licenses are enabled for the SAP system.

Download and configure the SAP libraries

To access SAP data through the Intermediate Documents (IDocs) or BAPI/RFC interface, 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 SAP 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:

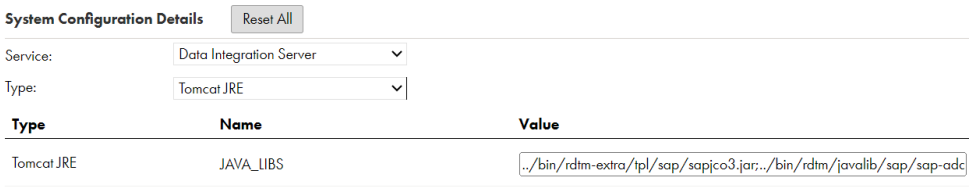
Secure Agent System	SAP JCo Libraries
Windows	`sapjco3.jar` `sapjco3.dll`
Linux	`sapjco3.jar` `libsapjco3.so`

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 enable data exchange through Intermediate Documents (IDocs) and BAPI/RFC interfaces.

You need to add the authorization object in SAP to process IDocs and SAP BAPI/RFC functions that helps you interact with SAP at run time. You also need access to specific IDoc and SAP BAPI/RFC functions underlying transactions that you want to process.

The following table describes the required authorization to process IDocs and SAP BAPI/RFC functions:

Function	Object Name	Authorization
SAP BAPI/RFC	S_RFC	SYST, SDTX, SDIFRUNTIME, RFC_METADATA, RFC1, RFC2, ABAP4_COMMIT_WORK, BAPI_TRANSACTION_COMMIT
IDocs	S_RFC	SYST, SDTX, SDIFRUNTIME, RFC1, RFC2, EDIMEXT

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

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.

You need the `sapnwrfc.ini` file to process SAP IDocs or SAP BAPI/RFC functions to facilitate transfer of data when you read from or write to SAP through the SAP IDoc or SAP BAPI/RFC interface.

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

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 create a connection, the Secure Agent first verifies if an `sapnwrfc.ini` file exists in the directory. 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\
```

For the serverless runtime environment, you need to copy the `sapnwrfc.ini` file to the following serverless agent directory:

```
\data2\home\cldagnt\SystemAgent\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

You can use the `sapnwrfc.ini` file to configure the following types of connections:

Connection to an 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.
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>
```

Define SAP Connector as a logical system in SAP

To receive IDocs from and send IDocs to SAP, you need to define SAP Connector as an external logical system in SAP.

To define SAP Connector as an external logical system, create a single logical system in SAP for IDoc ALE integration with SAP Connector, and then create an RFC destination configured with a tRFC port in the SAP system to communicate with SAP Connector.

To identify the external logical system, you also need to create a partner profile for the logical system that you created.

When you define SAP Connector as a logical system, SAP acknowledges SAP Connector as an external system that can receive outbound IDocs from SAP and send inbound IDocs to SAP.

Note: These procedures apply on the SAP version 4.6C. If you use a different version, the procedures may differ. For more information about how to create a logical system in SAP, see the SAP documentation.

Create a logical system for SAP Connector

You need to define SAP Connector as an external logical system in SAP to uniquely identify SAP Connector as a client within a network.

1. Log in to SAP and go to the SALE transaction.
2. On the **Display IMG** window, expand the tree to navigate to the **Application Link Enabling** > **Sending and Receiving Systems** > **Logical Systems** > **Define Logical System** operation.
3. Click the **IMG - Activity** icon to run the **Define Logical System** operation.
   An informational dialog box appears.
4. Click **Enter**.
   The **Change View Logical Systems** window appears.
5. Click **New Entries**.
6. On the **New Entries** window, enter a name and description for the logical system entry for SAP Connector.

Create an RFC destination

After you create a logical system, you need to create an RFC destination and program ID for SAP Connector in the SAP system.

1. Go to the SM59 transaction.
2. On the **Display and Maintain RFC Destinations** window, click **Create**.
   The **RFC Destination** window appears.
3. Enter the name of the logical system you created as the RFC destination.
4. To create a TCP/IP connection, enter T as the connection type.

5. Enter a description for the RFC destination.
6. Click **Save**.
7. For Activation Type, click **Registration**.
8. For Program ID, enter the same name as the RFC destination name.
   Use the Program ID as the value for the PROGRAM_ID parameter in the `sapnwrfc.ini` file.

Create a tRFC port for the RFC destination

After you create an RFC destination and program ID for SAP Connector, you need to create a tRFC port for the RFC destination you defined in SAP. SAP uses the tRFC port to communicate with SAP Connector.

1. Go to the WE21 transaction.
2. Click **Ports** > **Transactional RFC**.
3. Click **Create**.
   The **Ports in IDoc Processing** dialog box appears.
4. Click **Generate Port Name** or **Own Port Name** and enter a name.
5. Click **Enter**.
6. Enter a description for the port.
7. Select the IDoc record version type.
8. Enter the name of the RFC destination you created.

Create a partner profile for SAP Connector

Create a partner profile for the logical system you defined for SAP Connector. When SAP communicates with an external system, it uses the partner profile to identify the external system.

1. Go to the WE20 transaction.
2. Click **Create**.
3. Enter the following properties:

Partner Profile Property	Description
Partner number	Name of the logical system you created for SAP Connector.
Partner type	Partner profile type. Enter LS for logical system for ALE distribution systems.

4. In the **Post-processing** tab, enter the following properties:

Partner Profile Property	Description
Type	User type. Enter US for user.
Agent	The SAP user login name.
Lang	Language code that corresponds to the SAP language. Enter EN for English.

5. In the **Classification** tab, enter the following properties:

Partner Profile Property	Description
Partner class	Enter ALE.
Partner status	Indicates the status of communication with the partner. To communicate with the partner, enter A for active.

Create outbound and inbound parameters for the partner profile

After you define a partner profile for SAP Connector, you need to create outbound and inbound parameters for the partner profile.

Outbound parameters define the IDoc message type, IDoc basic type, and port number for outbound IDocs. Inbound parameters define the IDoc message type for inbound IDocs.

SAP uses outbound parameters when it sends IDocs to SAP Connector. Create an outbound parameter for each IDoc message type that SAP sends to SAP Connector. SAP uses inbound parameters when it receives IDocs from SAP Connector. Create an inbound parameter for each IDoc message type that SAP receives from SAP Connector.

1. From the **Partner Profiles** window in SAP, click **Create Outbound Parameter**.
2. On the **Partner Profiles: Outbound Parameters** window, enter the following properties:

Outbound Parameter Property	Description
Message Type	The IDoc message type the SAP system sends to SAP Connector.
Receiver Port	The tRFC port number you defined.
IDoc Type	The IDoc basic type of the IDocs the SAP system sends to SAP Connector.

3. Click **Save**.
   The **Packet Size** property appears.
4. Enter a value between 10 and 200 IDocs as the packet size.
   The packet size determines the number of IDocs that SAP sends in one packet to SAP Connector.
5. Click **Enter**.
6. Repeat steps from 1 to 5 to create an outbound parameter for each IDoc message type that SAP sends to SAP Connector.
7. Click **Create Inbound Parameter**.
8. On the **Partner Profiles: Inbound Parameters** window, enter the following properties:

Inbound Parameter Property	Description
Message Type	The IDoc message type the SAP system receives from SAP Connector.
Process Code	The process code. The SAP system uses the process code to call the appropriate function module to process the IDocs it receives.

9. Click **Enter**.
10. Repeat steps 7 through 9 to create an inbound parameter for each IDoc message type that SAP receives from SAP Connector.

Connect to SAP

Let's configure the SAP connection properties to connect to SAP through the Intermediate Documents (IDocs) or BAPI/RFC interface.

Before you begin

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

Check out "Prerequisites" on page 9 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
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 19.

SAP connection types

You can configure IDoc Reader, IDoc Writer, and SAP RFC/BAPI Interface connection types to access data in SAP through Intermediate Documents (IDocs) and BAPI/RFC interfaces.
Select the required SAP connection type and then configure the connection-specific parameters.

IDoc Reader connection

To read SAP data through the IDoc interface, select the **IDoc Reader** connection type and configure the connection properties.

The following table describes the basic connection properties for IDoc Reader connection:

Connection property	Description
Destination Entry	The DEST entry of the SAP application server specified in the `sapnwrfc.ini` file. Verify that the Program ID for this destination entry and the Program ID for the logical system you defined in the SAP system to receive IDocs are the same. For more information about the sapnwrfc.ini file, see "Configure the sapnwrfc.ini file" on page 11.
Code Page	The code page of the SAP application server defined in the connection when you read from IDocs. Select one of the following code pages from the list: - MS Windows Latin 1. Select for ISO 8859-1 Western European data. - UTF-8. Select for Unicode data. - Shift-JIS. Select for double-byte character data. - ISO 8859-15 Latin 9 (Western European). - ISO 8859-2 Eastern European. - ISO 8859-3 Southeast European. - ISO 8859-5 Cyrillic. - ISO 8859-9 Latin 5 (Turkish). - IBM EBCDIC International Latin-1.

IDoc Writer connection

To write SAP data through the IDoc interface, select the **IDoc Writer** connection type and configure the connection properties.

The following table describes the basic connection properties for IDoc Writer connection:

Property	Description
Username	The user name with the appropriate user authorization to connect to the SAP account.
Password	The password to connect to the SAP account.
Connection String	The DEST entry of the SAP application server specified in the `sapnwrfc.ini` file. For more information about the sapnwrfc.ini file, see "Configure the sapnwrfc.ini file" on page 11.
Code Page	The code page of the SAP application server defined in the connection when you write to IDocs. Select one of the following code pages from the list: - MS Windows Latin 1. Select for ISO 8859-1 Western European data. - UTF-8. Select for Unicode data. - Shift-JIS. Select for double-byte character data. - ISO 8859-15 Latin 9 (Western European). - ISO 8859-2 Eastern European. - ISO 8859-3 Southeast European. - ISO 8859-5 Cyrillic. - ISO 8859-9 Latin 5 (Turkish). - IBM EBCDIC International Latin-1.
Language Code	Language code that corresponds to the SAP language. Get the required language code from the SAP system to which you want to connect.
Client Code	The client number of the SAP application server. Get the required client number from the SAP system to which you want to connect.

SAP RFC/BAPI Interface connection

To access SAP data through the SAP RFC/BAPI interface, select the **SAP RFC/BAPI Interface** connection type and configure the connection properties.

The following table describes the basic connection properties for SAP RFC/BAPI Interface connection:

Property	Description
Username	The user name with the appropriate user authorization to connect to the SAP account.
Password	The password to connect to the SAP account.
Connection String	The DEST entry of the SAP application server specified in the `sapnwrfc.ini` file. For more information about the sapnwrfc.ini file, see "Configure the sapnwrfc.ini file" on page 11.
Code Page	The code page of the SAP application server defined in the connection when you read from or write to SAP BAPI/RFC functions. Select one of the following code pages from the list: - MS Windows Latin 1. Select for ISO 8859-1 Western European data. - UTF-8. Select for Unicode data. - Shift-JIS. Select for double-byte character data. - ISO 8859-15 Latin 9 (Western European). - ISO 8859-2 Eastern European. - ISO 8859-3 Southeast European. - ISO 8859-5 Cyrillic. - ISO 8859-9 Latin 5 (Turkish). - IBM EBCDIC International Latin-1.
Language Code	Language code that corresponds to the SAP language. Get the required language code from the SAP system to which you want to connect.
Client Code	The client number of the SAP server. Get the required client number from the SAP system to which you want to connect.

Use the serverless runtime environment

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

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

Before you configure an SAP 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 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:
        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.

CHAPTER 3

Data integration using BAPI/RFC functions

You can import a BAPI/RFC function as a mapplet to Data Integration. You can then use the mapplet in a mapping to read or write data by using the SAP RFC/BAPI Interface connection.

You can also use the BAPI/RFC Interface connection to read data from or write data to an SAP BAPI object that contains a hierarchical structure.

When you run a mapping or mapping task, Data Integration makes the RFC function calls to SAP to process data synchronously.

Business Application Programming Interfaces (BAPI) provide a way for third-party applications to synchronously integrate with SAP at the object-level. You use BAPIs to read, create, change, or delete data in SAP.

BAPIs allow access to the SAP system objects through methods for the business object types. Together with the business object types, BAPIs define and document the interface standard at the business level.

You define BAPIs in the SAP Business Objects Repository. You can call BAPIs as an ABAP program within SAP or from any external application. SAP Connector uses RFC protocol to call BAPI/RFC functions outside of SAP.

You can view and test the BAPI interface definitions in SAP using transaction SE37.

BAPI/RFC mapplets

You can generate SAP BAPI/RFC mapplets in Data Integration.

The BAPI/RFC mapplet includes a BAPI/RFC transformation. The BAPI/RFC transformation makes BAPI/RFC calls in SAP. BAPI/RFC calls include requests to the SAP system, such as creating, changing, or deleting data in SAP applications. To perform these tasks, BAPI/RFC functions use function parameter values.

Configuring SAP BAPI Connector as a Business Service

You can configure SAP BAPI Connector as a business service within a mapping or a mapping task for SAP BAPI data integration.

For more information, see How to configure SAP BAPI Connector as a business service Informatica How-To Library article.

BAPI/RFC mapplet parameters

BAPI/RFC functions use function parameter values to perform tasks. A BAPI/RFC mapplet includes input and output groups based on the BAPI/RFC transformation.

Function Parameters

BAPI/RFC functions can have the following parameters:

Function Parameter	Description
Scalar input parameters.	Scalar input values. Some BAPI functions require scalar input values to perform tasks such as changing data.
Scalar output parameters.	Scalar output values that a BAPI function returns after performing a task.
Table parameters.	SAP structures with more than one row. Table parameters can be input, output, or both. Input table parameters pass table input values to a BAPI/RFC function. For example, some BAPI/RFC functions require table inputs to change data.

Input and Output Groups

The BAPI/RFC transformation can contain the following groups:

Group Name	Description
Scalar input	Input group for scalar parameters. Contains a field for each scalar input parameter. The group name is SCALAR_INPUT and the field names are SI_<FieldName>.
Table input	One group for each table parameter. The fields represent import structures. The group name is TABLE_INPUT_<structure name> and field names are TI_<StructureName>_<FieldName>.
Scalar output	Output group for scalar parameters. Contains a field for each scalar output parameter. The group name is SCALAR_OUTPUT and field names are SO_<FieldName>.
Table output	One group for each table parameter. The fields represent output structures. The group name is TABLE_OUTPUT_<structure name> and field names are TO_<StructureName>_<FieldName>.
Error output	Passes data from data conversion errors and invalid BAPI/RFC calls. Map the error output field to a target to see error messages about data conversion and BAPI/RFC calls.

BAPI/RFC parameter properties

When you generate a BAPI/RFC mapplet, you can view the parameter properties and return structures.

You can also change the direction of the BAPI/RFC table parameters. The direction of the table parameters determine which groups in the transformation are input groups and which are output groups.

If the BAPI return structure is custom, you can edit the return structure properties that you selected during import for reusable BAPI/RFC transformations. Otherwise, you can view the BAPI/RFC return structure parameters.

The following table describes the BAPI/RFC function parameter properties:

Parameter	Description
Name	Name of the export, import, and table parameters and columns.
Associated Type	Definition of the parameter in SAP.
Short Description	Short description of the export, import, and table parameters and columns.
Optional	Indicates if the Secure Agent should pass a value to the parameter when it calls the BAPI.
Direction	Indicates if the parameter requires input or provides output. Values are: Input, Output, Both, or None.
Datatype	Object data type.
Precision	Object precision.
Scale	Object scale.
Default	Default value of the parameter in SAP, if any. SAP uses the default value when the Secure Agent does not pass a value to SAP.

The following table describes the parameter properties in the Return Structure tab:

Return Structure Parameter	Description
Return Structure	Return parameter name to determine the status of function calls. Value is RETURN if the BAPI contains a default return structure. If the BAPI does not contain a default return structure, select any table output parameter or scalar output parameter of type STRUCTURE. Default is None.
Status Field	Required if you select a value for the return structure. Select a field from the structure for status. If you select None for the return structure, this parameter is blank.
Text Field	Required if you select a value for the return structure. Select a field from the structure for status messages. If you select None for the return structure, this parameter is blank.
Status Indicator For Warning	Enter an indicator message for warning. If you select None for the return structure, the value is W. Default is W.
Status Indicator for Error	Enter an indicator message for error. If you select None for the return structure, the value is E. Default is E.
Status Indicator for Abort	Enter an indicator message for abort. If you select None for the return structure, the value is A. Default is A.

BAPI/RFC functions with nested structures

You can import metadata for a BAPI/RFC functions with nested structures.

A BAPI/RFC transformation includes multiple groups. When a BAPI function contains a nested structure, ports for the input and output groups in a BAPI/RFC transformation use the following naming convention:

```
<group_name>_<parameter_name>_<field name>
```

For example:

```
SCALAR_INPUT_PARAM1_FIELD1
```

If there are multiple input or output structures, the BAPI/RFC transformation includes each structure parameter name in the port names. For example, BAPI Z_TST2 has the parameter INPUT1, which is of the type ZTYPE1. ZTYPE1 has several components such as FIELD1 and FIELD2. FIELD2 is a component of the type structure. It contains field F1. The naming convention in the BAPI/RFC transformation for FIELD 1 is:

```
SCALAR_INPUT_INPUT1_FIELD1
```

The naming convention in the BAPI/RFC transformation for the field F1 is:

```
SCALAR_INPUT_INPUT1_FIELD2_F1
```

System variables

SAP uses system variables to set default values for some BAPI import parameters. The variables provide information, such as current date and time for the operating system on which SAP runs. System variables start with “SY-”. For example, SY-DATLO represents the local date of the SAP system.

The Secure Agent provides values for some system variables to define default input values for BAPI/RFC parameters. The Secure Agent uses the values as default input values for some ports of BAPI/RFC transformations. The Secure Agent uses the default values when there is no input for a port or when the port is not connected to an upstream transformation or source.

You can use the following system variables:

System Variable Name	Description
SY-LANGU	Log in language from the SAP application connection properties.
SY-MODNO	RFC handle value.
SY-MANDT	Value taken from the SAP application connection properties.
SY-DATUM	Local date on the Security Agent machine processing the data.
SY-UZEIT	Local time on the Security Agent machine processing the data.
SY-UNAME	Logon user ID from the SAP application connection properties.
SY-HOST	SAP host name from the SAP application connection properties.

Integration ID in BAPI/RFC mapplet

The Integration ID field is a key field in the BAPI mapplet. Each BAPI/RFC mapplet includes an Integration ID input field and output field.

When you run a mapping with a BAPI/RFC mapplet, the Secure Agent makes a BAPI/RFC call to SAP to process the data. By default, the BAPI/RFC mapplet commits every 1000 rows when you run a mapping.

The BAPI/RFC call is based on the input data of the Integration ID ports. The Secure Agent makes one call to SAP for each Integration ID. Pass a value to the Integration ID ports in the scalar input group and all mandatory table input groups of the BAPI/RFC mapplet.

Note: You must map the Integration ID input field even when a BAPI or RFC does not require other input fields.

If the BAPI/RFC call fails or if there is a data conversion error, SAP passes the data for the integration ID in comma-separated format to an error output group. If the mapping contains a target instance that is connected to the error output group, the Secure Agent writes the data to the target.

Target object for BAPI/RFC error output

To receive input data from a BAPI/RFC function call or data conversion errors from SAP, you can map a target transformation to a BAPI/RFC mapplet.

Create a target transformation with a column of the String data type and precision of 65535. Connect the column in the target object to the Error Output Group in the BAPI/RFC mapplet. The Secure Agent writes the error output data up to 65,535 characters to the target in comma-delimited format. If the error output data is longer than 65,535 characters, the Secure Agent truncates the data.

Rules and guidelines for BAPI/RFC mapplets in mappings

When you configure a mapping with a BAPI/RFC mapplet, use the following rules and guidelines:

- Pass a value to the Integration ID ports in the scalar input group and all mandatory table input groups of the BAPI/RFC transformation.
- Add a target object if you want to receive BAPI/RFC function call errors from the BAPI error group.

Use the following guidelines when passing data to BAPI/RFC function input parameters:

- When the function input parameter data type is INT1 or NUMC, provide positive values for the function input.
- When the source input data for a BAPI/RFC function is of the integer data type, do not use string data in the source transformation. Otherwise, the mapping fails.
- If the input data for a BAPI/RFC function mapping contains a higher scale than the SAP metadata specification, the Secure Agent rounds the data to comply with the SAP metadata. When you run a mapping in high precision mode, the mapping can fail due to overflow if the round-off data cascades to the precision digits. For example, the data type and precision for a BAPI/RFC function parameter is DEC (6,5). The input data that you pass to the function parameter is 9.99999. When the Secure Agent processes the input data, it rounds the input data to 10, which is not compatible with the SAP metadata. The mapping fails.
- When you run an SAP BAPI 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.

Mapping and mapping tasks using BAPI/RFC functions

You can generate a BAPI/RFC mapplet in Data Integration. You can then configure a mapping with the mapplet to manage data in SAP systems.

For example, to update sales order data in SAP, generate a BAPI/RFC mapplet from the BAPI_SALESORDER_CHANGE function and configure a mapping using the mapplet.

You can configure a mapping with a BAPI/RFC mapplet to pass input data to BAPI/RFC function input parameters.

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

To access the BAPI/RFC functionality through Data Integration, perform the following tasks:

1. Generate a BAPI/RFC mapplet in Data Integration.
2. Configure a mapping using the generated BAPI/RFC mapplet. Map the Integration ID field and other inputs and outputs. Save and run the mapping or create a mapping task using this mapping.

Configuring SAP BAPI Connector as a Business Service

You can configure SAP BAPI Connector as a business service within a mapping or a mapping task for SAP BAPI data integration.

For more information about configuring SAP BAPI Connector as a business service within a mapping or a mapping task, see the How-To Library article,
How to Configure SAP BAPI Connector as a Business Service in Cloud Data Integration.

Generating a BAPI/RFC mapplet

You can generate a BAPI/RFC mapplet in Data Integration and use the mapplet in a mapping.

Perform the following steps to generate a BAPI/RFC mapplet:

1. To create a mapplet, click **Data Integration** > **New** > **Mapplets** > **Mapplet - SAP BAPI** and then click **Create**.

   To edit a mapplet, on the **Explore** page, navigate to the mapplet. In the row that contains the mapplet, click **Actions** and select **Edit**.
2. Enter a unique name for the BAPI/RFC mapplet.
3. Optionally, enter a description for the BAPI/RFC mapplet that you want to generate.
4. Select the transformation type as **BAPI**.
5. Select the BAPI/RFC connection that you want to use.
6. Select the scope of the transformation.

   For real-time processing, select the transformation scope as **Transaction**. For batch processing and validation in a non-production environment, select the transformation scope as **All Input**.
7. Enter the name of the BAPI/RFC function.

   You can also search for a BAPI/RFC function by entering `%` as a wildcard character along with the search string. For example, to search for all BAPI/RFC functions whose names start with the term `Customer`, enter the search string as `Customer%`. To search for all BAPI/RFC functions whose names end with the term `Customer`, enter the search string as `%Customer`.
8. Click **Get Objects**.

   The **Select Source Object** dialog box appears displaying the search results. You can drill down the search results further by entering a search string in the **Select Source Object** dialog box.
9. Click the required BAPI/RFC function name and click **Select**.

   The import, export, and table parameter details of the BAPI/RFC function appear.
10. Specify the direction for the tables to indicate if the table parameters in the BAPI are input, ouput, or both.
11. Click **Save**.

    The mapplet for the specified BAPI/RFC function is created. You can create a mapping for the generated mapplet.

Configuring a mapping with a BAPI/RFC mapplet

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, you can enter a name and description.
5. Click the **Source** tab and configure the source details.

   Source details and advanced source properties appear based on the connection type.
6. To add a BAPI/RFC mapplet transformation, on the **Transformation palette**, click Mapplet.
   a. On the **General** tab, enter a name and description for the mapplet.
   b. Draw a link to connect the previous transformation to the mapplet transformation.
   c. On the **Mapplet** tab, click **Select**.

      The **Select Mapplet** dialog box appears.
   d. Specify a BAPI/RFC mapplet that you generated in Data Integration and click **OK**.
   e. Click **Connection** to specify an SAP RFC/BAPI Interface connection. You can create a connection, select a connection, or specify a parameter name for the connection.
   f. To preview fields, configure the field rules, or rename fields, click **Incoming Fields**.
   g. Click **Field Mapping** and map the incoming source fields with the Integration ID in the BAPI/RFC mapplet.
7. To add any other transformation, on the **Transformation palette**, click the transformation name. Or, drag the transformation onto the mapping canvas.
   a. On the **General** tab, 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.
8. 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.
   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.
9. Save and run the mapping or save and create a mapping task.

For more information about transformations, see *Transformations*.

Mappings with BAPI/RFC function example

You can use the `bapi_salesorder_createfromdat1` BAPI function to create sales order details for a customer in your organization.

In this example to create a sales order, perform the following tasks:

Step1: Generate a BAPI/RFC mapplet in Data Integration for the bapi_salesorder_createfromdat1 BAPI function.

Log in to Data Integration and generate a BAPI/RFC mapplet that contains the metadata of the bapi_salesorder_createfromdat1 BAPI function.

Step: 2 Configure a mapping using the generated mapplet.

Perform the following steps to configure a mapping:

1. Specify source objects to enter the order header data from the ORDER_HEADER_IN structure, the partner data using the ORDER_PARTNERS table, and item data using the ORDER_ITEMS_IN table as input parameters.
2. Add the Mapplet transformation. Draw a link to connect the flat file Source transformation to the Mapplet transformation. Draw the following links:
   - ORDER_HEADER_IN source object to the Scalar_Input input port of the BAPI mapplet.
   - ORDER_ITEMS_IN source object to the Table_Input_Order_Items_IN input port of the BAPI mapplet.
   - ORDER_PARTNERS source object to the Table_Input_Order_partners input port of the BAPI mapplet.
3. Configure the mapplet transformation.
   a. Select the generated mapplet. Verify that you specify an SAP RFC/BAPI Interface connection for the mapplet.
   b. Map the incoming source fields with the BAPI parameter properties.
4. Configure a flat file object to which you can write the sales order details. Draw a link to connect the Table Output in the Mapplet transformation to the flat file Target transformation. Create multiple flat file target objects to write the sales order. Create the following target objects and map them to the associated output ports in the BAPI mapplet:
   - ORDER_CFGS_BLOB, ORDER_CFGS_INST, ORDER_CFGS_PART_OF, ORDER_CFGS_REF, and ORDER_CFGS_VALUE to write item configuration data.
   - ORDER_ITEMS_OUT to write detailed item data.
   - ORDER_CCARD to write the credit card details.
   - ORDER_SCHEDULE_EX to write the structure of VBEP with English field names.
   - Scalar_Output to write the Scalar_Output from the mapplet.
   - Error_Output to write the Error_Output from the mapplet.

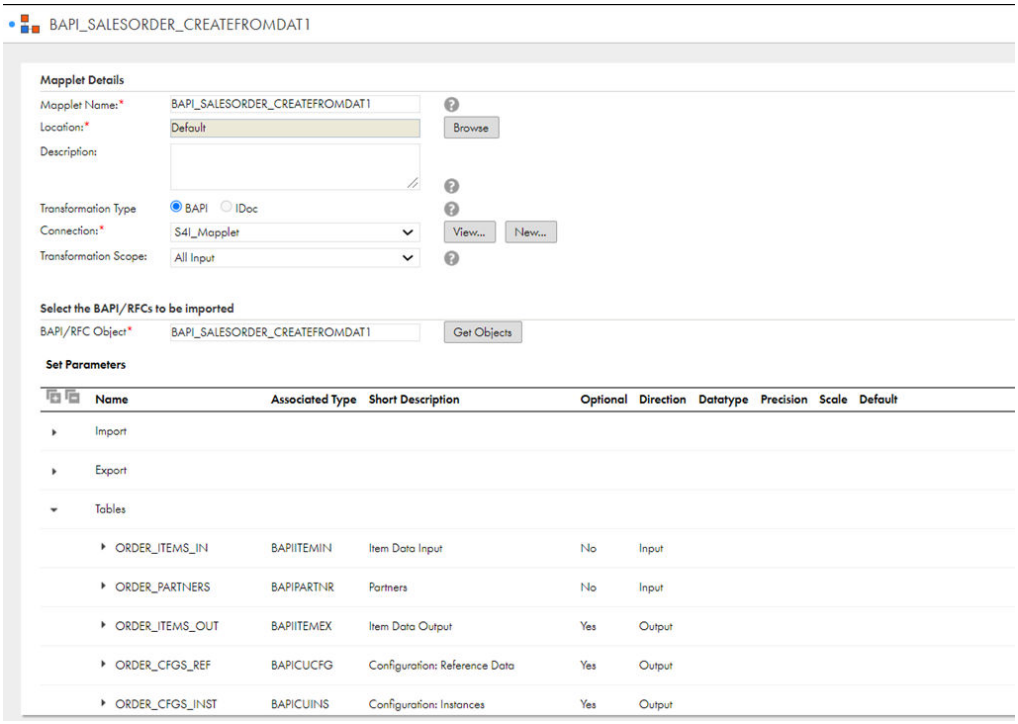
Step 1: Generating a BAPI/RFC mapplet in Data Integration for the BAPI_SALESORDER_CREATEFROMDAT1 BAPI function

Perform the following steps to generate a BAPI/RFC mapplet:

1. To create a mapplet, click **Data Integration** > **New** > **Mapplets** > **Mapplet - SAP** and then click **Create**.

   To edit a mapplet, on the **Explore** page, navigate to the mapplet. In the row that contains the mapplet, click **Actions** and select **Edit**.

   The **New SAP Mapplet** page appears. The following image shows the **New SAP Mapplet** page:



2. Enter a unique name for the BAPI/RFC mapplet.
3. Optionally, enter a description for the BAPI/RFC mapplet you want to generate.
4. Select the transformation type as **BAPI**.
5. Select the BAPI/RFC connection that you want to use.
6. Select the scope of the transformation.

   In real time, select the transformation option as Transaction. For batch processing and validating in non-production environment, select ALL INPUT as the transformation scope.
7. Enter the name of the BAPI/RFC function.

   Note: Verify that you enter the exact name of the BAPI/RFC function that you want to retrieve.
8. Click **Get Objects**.

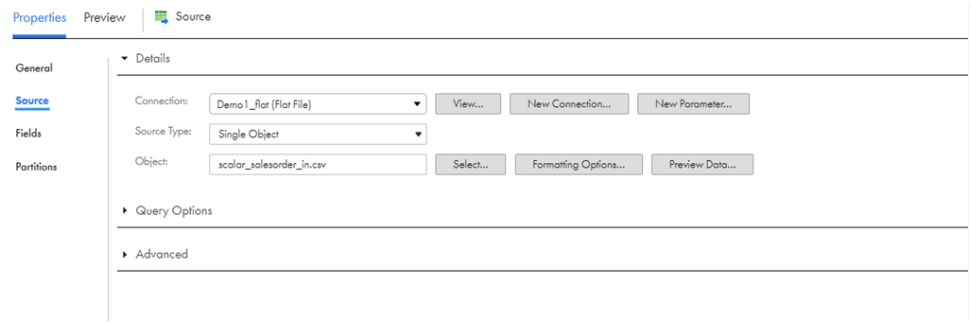
   You can view the import, export, and table parameter details of the BAPI/RFC function.
9. Specify the direction for the tables to indicate if the table parameters in the BAPI are input, ouput, or both.
10. Click **OK**.

    The mapplet for the specified BAPI/RFC function is created. You can create a mapping for the generated mapplet.

Step 2: Configuring a mapping with the bapi_salesorder_createfromdat1 mapplet

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**.
3. To configure the flat file 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 the source details.
   a. Select a flat file connection.
   b. Select Single Object as the **Source Type**.
   c. Click **Select** to specify a flat file that contains the source fields.

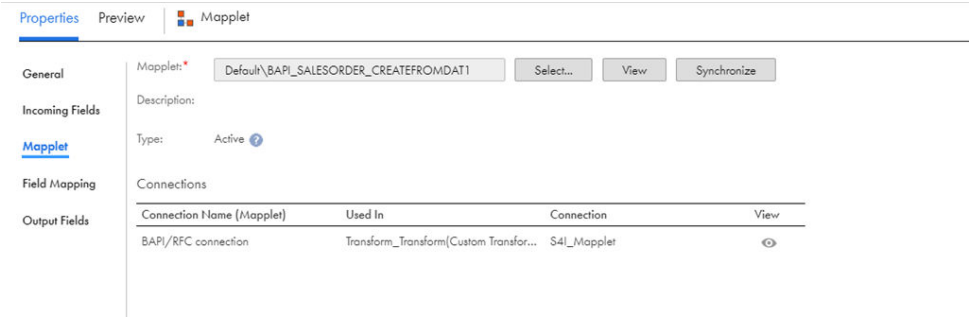
      The following image shows the flat file source details:



6. Add order header, order items, and order partners as flat file source objects for the mapping.
7. To add a BAPI/RFC mapplet transformation, on the **Transformation palette**, click Mapplet.
   a. On the **General** tab, enter a name and description for the mapplet.
   b. Draw a link to connect the flat file Source transformation to the Mapplet transformation.
   c. On the **Mapplet** tab, click **Select**.

      The **Select Mapplet** dialog box appears.
   d. Select the `bapi_salesorder_createfromdat1` mapplet and click **OK**.
   e. Click **Connection** to specify an SAP RFC/BAPI Interface connection.

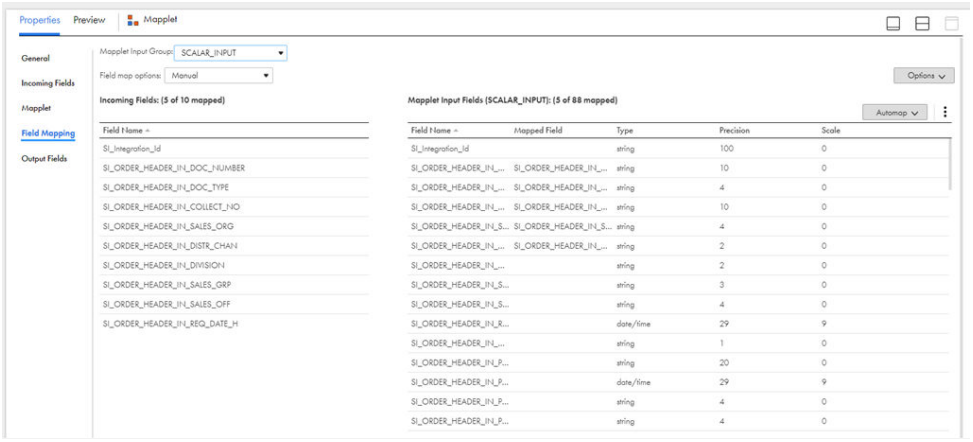
      The following image shows the bapi_salesorder_createfromdat1 mapplet details:



   f. To preview fields, configure the field rules, or rename fields, click **Incoming Fields**.

g. Click **Field Mapping** and map the incoming source fields with the appropriate mapplet input fields.

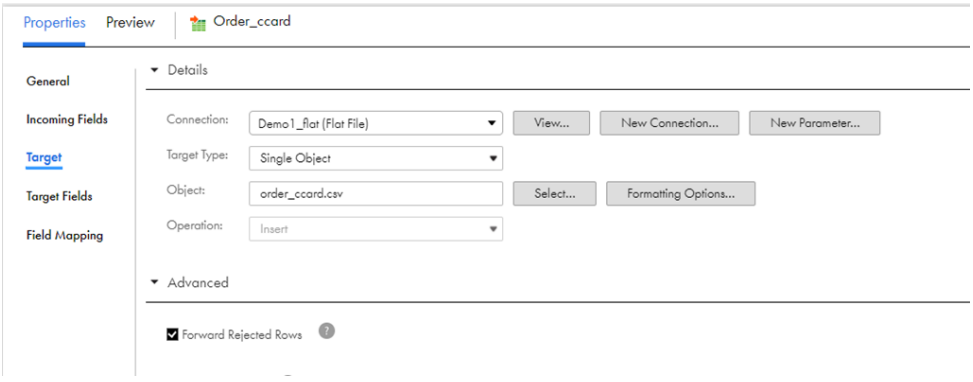
The following image shows the field mapping of the incoming source fields with the mapplet input fields:



8. To add a flat file target, on the **Transformation palette**, click **Target**.
9. On the **General** tab, enter a name and description for the target.
10. Click the **Target** tab and configure target details.
    a. Select a flat file connection for the target.
    b. Select Single Object as the **Source Type**.
    c. Click **Select** to specify the target object.

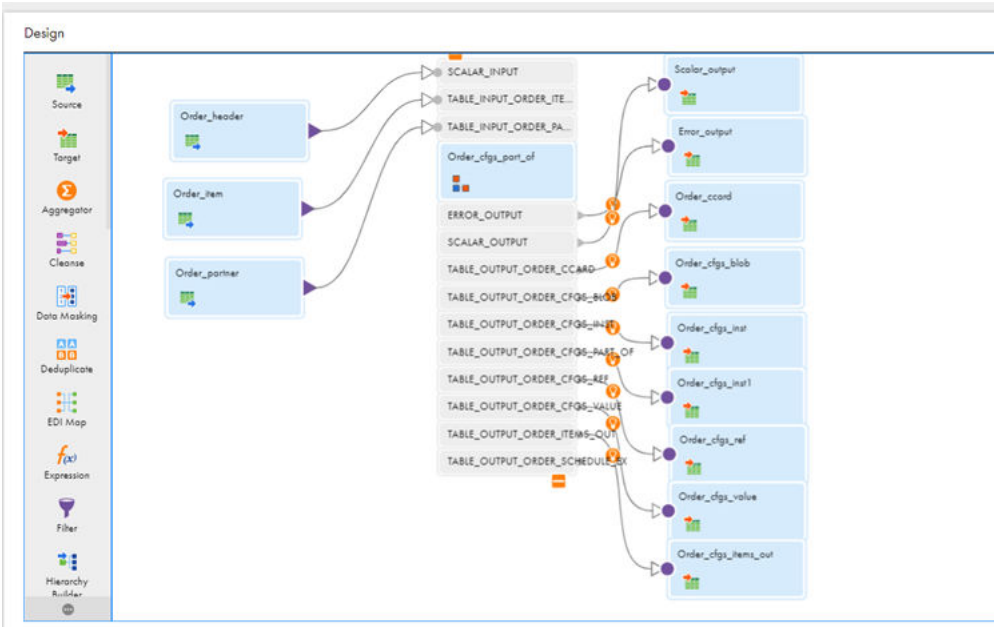
    The **Target Object** dialog box appears.
    d. Select the flat file object, and click **OK**.
    e. Optionally, you can choose to forward rejected rows in the Advanced Target options.
    f. Repeat the above steps to add all the flat file target objects.

    The following image shows the target transformation details:



11. Draw a link to connect the output ports in the Mapplet transformation to the flat file Target transformation.

    The following image shows the mapping configured for the bapi_salesorder_createfromdat1 mapplet:



12. Save and run the mapping.

SAP mapplet properties

You can commit data, cache data, and handle errors for the target in an SAP BAPI/RFC mapping task.

To configure the SAP mapplet properties for the target, click the **Runtime Options** tab of the SAP BAPI/RFC mapping task. Expand the **SAP Mapplet Properties** section, and configure the mapplet properties.

The following table describes the commit properties you can configure in an SAP BAPI/RFC mapping task:

Property	Description
Perform Commit	Issues a commit with a BAPI commit function. If you disable this option, Data Integration does not call a commit function. It also ignores the commit interval and does not wait for a commit to continue to run the mapping task. Default is enabled.
Perform Commit Wait	Waits until the commit is complete to continue to run the mapping task. SAP sends a return call to Data Integration when the commit is complete. If you disable this option, Data Integration continues to process data during the commit. Default is enabled.

Property	Description
Commit Interval	Number of BAPI/RFC calls after which Data Integration issues a commit. To issue a commit, Data Integration makes a BAPI/RFC commit call. If you disable the **Perform Commit** option, Data Integration ignores this value. Default is 1000.
Commit Function	Determines how Data Integration makes a BAPI/RFC call when it issues a commit. Data Integration makes the call either from outside SAP or from within BAPI based on the following options you select: - ABAP4_COMMIT_WORK. Makes a BAPI/RFC call from ABAP and then issues a commit. - BAPI_TRANSACTION_COMMIT. Makes call from outside SAP and then issues a commit. If you disable the **Perform Commit** option, Data Integration ignores this value. Default is BAPI_TRANSACTION_COMMIT.

The following table describes the data caching properties you can configure in an SAP BAPI/RFC mapping task:

Property	Description
Cache Directory	Default directory used to cache BAPI data. By default, the cache files are created in a directory specified by the variable $PMCacheDir. If you override the directory, make sure the directory exists and contains enough disk space for the cache files.
Cache Size	Total memory in bytes allocated to Data Integration to cache data prepared for BAPI/RFC mapplets. Default is 10 MB.

The following table describes the error handling properties you can configure in an SAP BAPI/RFC mapping task:

Property	Description
Duplicate Parent Row Handling	Determines how Data Integration handles duplicate parent rows in a mapping task. Select one of the following options: - First Row. Passes the first duplicate row to the target. Data Integration rejects rows with the same primary key that it processes after this row. - Last Row. Passes the last duplicate row to the target. - Error. Data Integration passes the first row to the target. Rows that follow with duplicate primary keys increment the error count. The mapping task fails when the error count exceeds the error threshold. Default is First Row.
Orphan Row Handling	Determines how Data Integration handles orphan rows in a mapping task. Select one of the following options: - Ignore. Ignores orphan rows. - Error. The mapping task fails when the error count exceeds the error threshold. Default is Ignore.
Continue on Error	Determines if you want to continue to run the mapping task when a BAPI/RFC error is encountered. Default is enabled.

CHAPTER 4

Data integration using IDocs

You can generate an IDoc mapplet to Data Integration. You can then use the mapplet in an outbound mapping to read IDocs from SAP or inbound mapping to write IDocs to SAP.

IDocs electronically exchange data between SAP applications or between SAP applications and external programs. IDoc is a message-based integration interface that processes data asynchronously.

IDoc is a component of Application Link Enabling (ALE) module within SAP that can send and receive Intermediate Documents (IDocs) over RFC protocol.

ALE Layers

The message-based architecture of ALE comprises three layers:

- Application layer that provides ALE an interface to SAP to send or receive messages from external systems.
- Distribution layer that filters and converts messages to ensure that they are compatible between different SAP releases.
- Communications layer that enables ALE to support synchronous and asynchronous communication. You use IDocs for asynchronous communication.

The architecture of ALE provides a way to send IDocs as text files without connecting to a central database. Applications can communicate with each other without converting between formats to accommodate hardware or platform differences.

IDoc Record Types

IDocs contain three record types:

- Control record, which identifies the message type.
- Data records that contain the IDoc data in segments.
- Status records that describe the status of the IDoc. Status record names are the same for each IDoc type.

ALE Components

ALE has the following components:

Component	Description
Logical System	All systems that need to communicate using ALE/IDoc must be setup as a Logical System within SAP. An SAP administrator can setup logical systems in transaction BD54.
Distribution Model	Defines an agreement between two logical systems on the messages that can be exchanged and identifies the sender and the receiver. An SAP administrator can setup distribution models in transaction BD64.

Component	Description
Partner Profile	Stores the IDoc type and processing logic related to the distribution model. An SAP administrator can setup partner profiles in transaction WE20.
RFC Destination	Defines the protocol and access to the logical system. An SAP administrator can setup RFC destinations in transaction SM59.
Message Type	Representation of a business object.
IDoc Type	Representation of a message type. SAP uses IDoc types to support backward compatibility across various SAP releases.
IDoc	An instance of an IDoc type that contains business data.

IDoc mapplets

You can import an IDoc as a mapplet using the **Mapplet - SAP IDoc** asset. An IDoc contains a hierarchical structure consisting of segments. Each segment is an SAP structure defined in the SAP system.

An IDoc has header and data record components. The header component contains control information, such as creation date and status. The control information is in an SAP structure called EDIDC. The data records are in an SAP structure called EDIDD.

Segments and groups

An IDoc is a hierarchical structure that contains segments. A segment can be a parent or child. A child segment depends on another segment. A parent segment contains child segments. A parent segment can be a child of another segment.

IDoc segments are organized into groups. The following rules determine the group to which a segment belongs:

- A parent segment starts a new group. For example, in the MATMAS04 IDoc, the E1MARCM segment contains a child and therefore starts a group.
- A child segment that is not a parent belongs to the group that is started by its immediate parent. For example, in the MATMAS04 IDoc, the E1MARA1 segment does not contain a child and therefore belongs to the group of its parent E1MARAM.
- A group can also be a parent or a child.

IDocs properties

When you fetch an IDoc in **Mapplet - SAP IDoc**, you can view all the IDoc segments. Select a segment to view the fields in the segment.

IDocs Properties

The following table describes the IDocs properties that you can view and specify in the **Mapplet - SAP IDoc** asset:

Property	Description
Transformation Type	Indicates the type of transformation you want to use to generate the IDoc mapplet. Select one of the following options: - IDoc Prepare. Select to generate a mapplet that writes source data as an IDoc message. - IDoc Interpreter. Select to generate a mapplet that reads IDoc messages.
Source	The source by which you fetch the IDoc. Select one of the following options: - Connection. Select to fetch the IDoc with an SAP Table Connector connection using the **Saprfc.ini Path** and **Destination** advanced properties. For more information about how to create an SAP Table connection, see the *SAP Table Connector* documentation. - Local file. Select to fetch the IDoc by using a file from your local drive. For more information about importing IDoc metadata for an SAP/ALE IDoc transformation from file, see "Generating IDoc metadata to local file" on page 37.
Transformation Scope	Indicates how the Secure Agent applies the transformation logic to incoming data. Select one of the following options: - Transaction. Select Transaction to apply the transformation logic to all rows in a transaction. Select Transaction when the results of the transformation depend on all rows in the same transaction, but not on rows in other transactions. When you select Transaction, associated mappings can run in real time. - All Input. Select All Input to apply the transformation logic to all incoming data, and to drop the incoming transaction boundaries. Select All Input when the results of the transformation depend on all rows of data in the source. Default is **All Input**.
Message Type	The type of application messages that classify categories of data. For example, ORDERS and MATMAS (Material Master).
Basic IDoc Type	The type of data structure associated with the message type. For example, MATMAS01, MATMAS02 for MATMAS. IDocs contain the data associated with the message type.
Extended Type	Extended IDoc type name. IDoc extension is extension of basic type and contains additional custom IDoc segments and fields that are not available in the standard basic type.
IDoc Document	Displays the IDoc segments associated with the message type and IDoc type.
Segment Name	Segment names of the IDoc type.
Description	Description of the segments.
Min Occurs	Minimum number of occurrences of the segment in an IDoc.

Property	Description
Max Occurs	Maximum number of occurrences of the segment in an IDoc.
Select	Selects the data segments to include in the transformation. When you select a segment, the parent segments and all required child segments are also selected. When you clear a segment, all child segments are also cleared.
Control Record	Displays the control record segments in the IDoc, their values, and precision. The Designer provides values for some of the segments when you create a mapplet using the IDoc prepare transformation. You can also provide values for other segments.

Segment Definition

You can view the segment field details from the segment definition section. To view the field names of the segment, select a segment name.

The following table describes the segment field details:

Field	Description
Name	Field name of a segment.
SAP Datatype	SAP data type of the field.
Precision	Precision of the field.
Scale	Scale of the field.
Description	Description of the field.

Generating IDoc metadata to local file

You can import IDoc metadata for an SAP/ALE IDoc transformation from a local file.

To import IDoc metadata from a local file, run the RSEIDoc3 program from the SAP client to generate the metadata. When you run the program, select the IDoc type and range for the IDoc metadata you want to generate. The program exports the metadata it generates into a metadata file. For example, you can export the metadata to a file with the *.idc* extension. You can then use the metadata file to import the metadata into the Designer for use in the SAP/ALE IDoc transformation.

To generate IDoc metadata using the RSEIDoc3 program:

1. Enter transaction se38 from the SAP client.
2. Execute the RSEIDoc3 program.
3. Select the basic IDoc type and range.
4. If you are using SAP version 5.0 or later, click **Control Record**.
5. Optionally, select extended IDoc type and range, if available.
6. Optionally, select extended grammar, if available.
7. Click **Parser**.
8. Click **System** > **List** > **Save** > **Local File**.
9. On the Save List in File dialog box, select **Unconverted**.

10. Enter the path and file name where you want to save the metadata file, and then save the file with the .idc extension.

Outbound mapplet

You can capture changes to the master data or transactional data in the SAP application database in real time.

When data in the application database changes, the SAP system creates IDocs to capture the changes and sends the IDocs to Data Integration. You can use the IDoc Reader connection to read the IDoc message in real time as they are generated by the SAP system.

If the Secure Agent is not running when the SAP system sends outbound IDocs, the Secure Agent does not receive the IDocs. However, the SAP system stores the outbound IDocs in EDI tables, which are a staging area for guaranteed message delivery. You can configure the SAP system to resend the IDocs by configuring the tRFC port used to communicate with the Secure Agent. When you configure the port, you can enable background processes in SAP that try to resend the IDocs to the Secure Agent a set number of times.

To generate the outbound mapplet to read IDoc messages from SAP system, use the IDoc Interpreter when you import the IDoc metadata. Import the outbound mapplet to Data Integration and configure an outbound mapping.

Outbound mapplet ports

An outbound IDoc mapplet contains predefined ports. You cannot edit the ports.

The following table describes the mapplet ports:

Port Name	Description
Basic IDoc Type	Basic IDoc type name. Basic IDoc type defines the structure of an IDoc. Each basic type describes standard IDoc segments, format of data fields, and size. Basic type contains all the standard fields that are necessary for carrying out a business transaction.
Extended IDoc Type	Extended IDoc type name. IDoc extension is extension of basic type and contains additional custom IDoc segments and fields that are not available in the standard basic type.
IDocRecord	IDoc message data.
DocumentNumber	Unique message number of the IDoc.

Target object for outbound mapplet error output

You can configure an outbound IDoc mapping to write IDocs that are not valid to a relational or flat file target. To write IDocs that are not valid to a relational or flat file target, connect the IDoc_Interpreter_Error_Output_Group port in the outbound mapplet to a relational or flat file target object. You must also configure the error log type session property in the **Schedule** page. Based on your requirement, you can choose to configure multiple target objects for each segment in the IDoc and for the IDoc_Interpreter_Error_Output_Group.

Inbound mapplet

You can synchronize transactional data in a legacy application with the data in the SAP application database.

Use an inbound SAP IDoc mapping to send the transactional data from the legacy application database to the SAP system. Data Integration extracts the data from the legacy application data source, prepares the data in SAP IDoc format, and sends the data to the SAP system as inbound IDocs using ALE. You can use the IDoc Writer connection to write inbound SAP IDoc messages to SAP systems.

To generate the inbound mapplet to write IDocs to SAP systems, use the Prepare transformations when you import the IDoc metadata. Import the inbound mapplet to Data Integration and configure an inbound mapping.

Key fields and control record fields

An IDoc mapplet includes a primary key (GPK) and a foreign key (GFK) in each segment. When you configure a mapping, integration template, or mapping task, map the primary key field, foreign key field, and any control record fields.

Note: You can enable control record fields when you import the IDoc metadata.

The Prepare transformation in the SAP IDoc Writer mapplet can have primary key and foreign key fields and other input fields, including control record fields.

The Prepare transformation has the following output fields:

- IDoc Data. Map this field to an IDoc target.
- Error IDoc Data. Map this field to see error messages about IDoc syntax/data conversion.

When you import IDoc metadata, you can add fields to an **IDoc Prepare** transformation. In **Mapplet - SAP IDoc**, on the **Control Record** tab, you can add the control record keys as key-value pairs. For example, you can add the following key-value pairs for the sender partner type and the sender partner number:

- Key: SNDPRT and Value: LS for Logical System
- Key: SNDPRN and Value ICS

In an inbound mapping, you can pass the sender partner number to SAP. You can pass a value to the CONTROL_SNDPRN port in the control input group of the Prepare transformation. If you do not connect this port to an upstream transformation, the Secure Agent uses the partner number value of SNDPRN key you specify in the **Mapplet - SAP IDoc** asset.

IDoc primary and foreign keys

An IDoc message is organized hierarchically with one top-level parent segment and one or more second-level child segments. Second-level child segments can also have one or more third-level child segments.

To maintain the structure of the IDoc data, the Prepare transformation in the SAP IDoc Writer mapplet uses primary and foreign keys. The top-level parent segment has a primary key. Each child segment has a primary key and a foreign key. The foreign key of each child segment references the primary key of its parent segment. For example, the foreign key of a second-level child segment references the primary key of the top-level parent segment. Similarly, the foreign key of a third-level child segment references the primary key of the second-level child segment.

The Prepare transformation groups incoming IDoc data based on the values in the primary and foreign key fields. The Control Input group of the Prepare transformation represents the parent segment. All other groups of the Prepare transformation except the ErrorIDocData group represent second-level or third-level child segments. The ErrorIDocData group is used for processing invalid IDocs.

The following table shows the groups of the Prepare transformation and the fields used for the primary and foreign keys:

Groups	Field	Description
Control Input Group	GPK_DOCNUM	Primary key of the parent segment.
Child Segment 1	GPK_<Child1_name>	Primary key of Child Segment 1.
Child Segment 1	GFK_DOCNUM_<Child1_name>	Foreign key of Child Segment 1 references the primary key of the parent segment.
Child Segment A of Child Segment 1	GPK_<Child1A_name>	Primary key of Child Segment A of Child Segment 1.
Child Segment A of Child Segment 1	GFK_<Child1_name>_<Child1A_name>	Foreign key of Child Segment A of Child Segment 1 references the primary key of Child Segment 1.
Child Segment 2	GPK_<Child2_name>	Primary key of the IDoc child segment.
Child Segment 2	GFK_DOCNUM_<Child2_name>	Foreign key of Child Segment 2 references the primary key of the parent segment.
Child Segment B of Child Segment 2	GPK_<Child2B_name>	Primary key of Child Segment B of Child Segment 2.
Child Segment B of Child Segment 2	GFK_<Child2_name>_<Child2B_name>	Foreign key of Child Segment B of Child Segment 2 references the primary key of Child Segment 2.

Each value for the GPK_<name> field needs to be unique. Each GFK_<parent_name>_<group_name> field needs to reference the primary key of its parent segment.

For example, the following table shows the relationship of primary and foreign keys in an IDoc message named ABSEN1 with four child segments:

Group	Field	Primary/Foreign Keys
CONTROL_INPUT_ABSEN1	GPK_DOCNUM	P1
E2ABSE1	GPK_E2ABSE1	C1
-	GFK_DOCNUM_E2ABSE1	P1
E2ABSE2	GPK_E2ABSE2	C2
-	GFK_DOCNUM_E2ABSE2	P1
E2ABSE2A	GPK_E2ABSE2A	C2A
-	GFK_E2ABSE2_E2ABSE2A	C2
E2ABSE3	GPK_E2ABSE3	C3
-	GFK_DOCNUM_E2ABSE3	P1
E2ABSE3B	GPK_E2ABSE3B	C3B

Group	Field	Primary/Foreign Keys
-	GFK_E2ABSE2_E2ABSE2A	C3
E2ABSE4	GPK_E2ABSE4	C4
-	GFK_DOCNUM_E2ABSE4	P1

The Prepare transformation uses these primary and foreign key relationships to maintain the structure of the IDoc data. Any foreign key field that does not match the primary key of its parent segment results in an orphan row. Any primary key field that is not unique results in a duplicate row.

Verify that each IDoc message has a unique primary key for the top-level parent segment, each child segment, and that each foreign key matches the primary key of its parent.

Configuring IDoc mapplets

You can configure IDoc mapplets using the **Mapplet - SAP IDoc** asset from Data Integration. You can use IDoc mapplets to the inbound or outbound mappings.

1. In Data Integration, click **New** > **Mapplets** > **Mapplet - SAP IDoc**, and then click **Create**.
2. Enter a unique name for the mapplet.
3. Specify the location where you save the mapplet.
4. Optionally, enter a description for the mapplet.
5. Select the transformation type. You can select one of the following options:
   - Select **IDoc Interpreter** to generate outbound mappings to read IDocs from the SAP system.
   - Select **IDoc Prepare** to generate inbound mappings to write IDocs to the SAP system.
6. Select the source by which you fetch the IDoc. You can select one of the following options:
   - Select **Connection** to fetch the IDoc with an SAP Table Connector connection using the **Saprfc.ini Path** and **Destination** advanced properties.

     Note: Ensure that you add the following entries in the `sapnwrfc.ini` file when you create the SAP Table connection:

     ```
     DEST=PRD
     ASHOST=sapr3
     SYSNR=01
     ```

     The SAP Table connection information is available in the SAP Table Connector documentation. For more information about the SAP Table connection, see the *SAP Table Connector* documentation.
   - Select **Local File** to fetch the IDoc using a file from your local drive.
7. Select the scope of the transformation. You can select one of the following options:
   - Select **Transaction** to apply the transformation logic to all rows in a transaction. Select **Transaction** when the results of the transformation depend on all rows in the same transaction, but not on rows in other transactions. When you select **Transaction**, associated mappings can run in real time. For outbound mappings, select **Transaction**.
   - Select **All Input** to apply the transformation logic to all incoming data, and to drop the incoming transaction boundaries. Select **All Input** when the results of the transformation depend on all rows of data in the source.
8. Enter the message type and the basic IDoc type, and then click **Fetch IDoc**.

The segment details of the IDoc appears.

9. On the **IDoc Document** tab, in the **Select IDoc Segments** section, select the segments you want to include in the mapplet.
10. Optionally, in an inbound mapping, to add a control record field to the mapplet, on the **Control Record** tab, select the checkbox for the field you want to add. This enables you to map the selected control record fields when you configure a mapping, integration template, or mapping task.
11. Click **Save**.

Mapping and mapping tasks using IDocs

To read and write IDocs, Data Integration integrates with SAP applications using Application Link Enabling (ALE).

ALE is an SAP proprietary technology that enables data communication between SAP systems. ALE also enables data communication between SAP and external systems.

You can configure outbound mappings to read IDocs from SAP and inbound mappings to write IDocs to SAP.

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

To configure an outbound mapping to read IDocs from SAP, perform the following tasks:

1. Configure an IDoc mapplet using the **Mapplet - SAP IDoc** asset from Data Integration. Verify that you selected the segments and groups you want to include in the IDoc. In addition, select the **IDoc Interpreter** transformation.
2. Configure an outbound mapping using the generated IDoc mapplet. Add a Source transformation to read data from the SAP system, add the generated IDoc mapplet, and add a Target transformation to write the IDoc in the target object.

To configure an inbound mapping to write IDocs to SAP, perform the following tasks:

1. Create an IDoc mapplet using the **Mapplet - SAP IDoc** asset from Data Integration. Verify that you selected the segments and groups you want to include in the IDoc. In addition, select the **IDoc Prepare** transformation.

   Note: Do not configure multiple IDoc Prepare transformations for an IDoc mapplet in an inbound mapping on Linux. Otherwise, the mapping fails with the DTM error.
2. Configure an inbound mapping using the generated IDoc mapplet. Add a Source transformation to read data from the source system, add the generated IDoc mapplet, and add a Target transformation to write the IDoc to the SAP system.

IDoc reader sources in mappings

To read IDocs from an SAP application, use an SAP IDoc Reader connection and configure the IDoc Reader source properties in the Source transformation in a mapping.

Specify the name and description of the IDoc Reader source. Configure the source and advanced properties for the source object.

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

Property	Description
Connection	Name of the source connection.
Source Type	Source type. Select **Single** for a single source object. When you select an SAP IDoc Reader as connection, the source type can be a single object and source object is the IDoc Reader Object. The source object has the generic structure of an IDoc message.
Object	Source object.

The following table describes the SAP IDoc Reader advanced source properties:

Property	Description
Idle Time	Indicates the number of seconds the Secure Agent waits for IDocs to arrive before it stops reading from the SAP source. For example, if you enter 30 seconds for idle time, the Secure Agent waits 30 seconds after reading from the SAP source. If no new IDocs arrive within 30 seconds, the Secure Agent stops reading from the SAP source. Default is 300.
Packet Count	Controls the number of packets the Secure Agent reads from SAP before stopping. For example, if you enter 10 for Packet Count, the Secure Agent reads the first 10 packets from the SAP source and then stops. The packet Size property in the ALE configuration determines the number of IDocs the Secure Agent receives in a packet. If you enter packet count as -1, you can read infinite number of packets. Default is -1.
Realtime Flush Latency	Determines, in seconds, how often the Secure Agent flushes data from the source.
Reader Time Limit	Sets a period of time, in seconds, during which the Secure Agent reads IDocs from the SAP source. For example, if you specify 10 as the reader time limit, the Secure Agent stops reading from the SAP source after 10 seconds. If you enter reader time limit as 0, the Security Agent continues to read IDocs from SAP for an infinite period of time. Default is 0.
Recovery Cache Folder	Specifies the location of the recovery cache folder. **Note:** This property is not applicable when you configure the SAP IDoc Reader Source transformation.
Number of Retries for Connection Resiliency	Determines the number of times Cloud Data Integration to attempt to connect to the SAP system. Default is 1.
Delay between Retries for Connection Resiliency	Determines the time interval in seconds between the connection retries to connect to the SAP system if it could not connect on a previous attempt. Default is 5.
Tracing Level	Sets the amount of detail that appears in the log file. You can choose terse, normal, verbose initialization or verbose data. Default is normal.

IDoc writer targets in mappings

You can use an SAP IDoc Writer connection and configure the IDoc Writer target properties in the Target transformation in a mapping.

Specify the name and description of the IDoc Writer target. Configure the target and advanced properties for the target object in a Target transformation.

The following table describes the IDoc Writer target properties that you can configure in a Target transformation:

Property	Description
Connection	Name of the target connection.
Target Type	Type of the target object.
Object	Name of the target object.

The following table describes the SAP IDoc Writer advanced target properties:

Property	Description
Packet Size	Number of IDocs that you want to send in a packet to SAP.
Number Of Retries	Not applicable.
Delay Between Retries	Not applicable.
Send IDocs Based On	The commit method of IDocs to SAP based on the value you set for the **Packet Size** property or commit properties at every commit point. Select one of the following options: - Packet Size. Commits IDocs to SAP based on the value you set for the **Packet Size** property. Data Integration collects IDoc messages until the total count reaches the packet size. It then sends messages as a packet to SAP. - Commit Call. Commits IDocs to SAP based on the commit properties at every commit point.
Generate Request ID	Not applicable.
Forward Rejected Rows	Determines whether the transformation passes rejected rows to the next transformation or drops rejected rows. By default, the agent forwards rejected rows to the next transformation.

Configuring IDoc mapplets

You can configure IDoc mapplets using the **Mapplet - SAP IDoc** asset from Data Integration. You can use IDoc mapplets to the inbound or outbound mappings.

1. In Data Integration, click **New** > **Mapplets** > **Mapplet - SAP IDoc**, and then click **Create**.
2. Enter a unique name for the mapplet.
3. Specify the location where you save the mapplet.
4. Optionally, enter a description for the mapplet.
5. Select the transformation type. You can select one of the following options:

- Select **IDoc Interpreter** to generate outbound mappings to read IDocs from the SAP system.
- Select **IDoc Prepare** to generate inbound mappings to write IDocs to the SAP system.

6. Select the source by which you fetch the IDoc. You can select one of the following options:
   - Select **Connection** to fetch the IDoc with an SAP Table Connector connection using the **Saprfc.ini Path** and **Destination** advanced properties.

     Note: Ensure that you add the following entries in the `sapnwrfc.ini` file when you create the SAP Table connection:

     ```
     DEST=PRD
     ASHOST=sapr3
     SYSNR=01
     ```

     The SAP Table connection information is available in the SAP Table Connector documentation. For more information about the SAP Table connection, see the *SAP Table Connector* documentation.
   - Select **Local File** to fetch the IDoc using a file from your local drive.
7. Select the scope of the transformation. You can select one of the following options:
   - Select **Transaction** to apply the transformation logic to all rows in a transaction. Select **Transaction** when the results of the transformation depend on all rows in the same transaction, but not on rows in other transactions. When you select **Transaction**, associated mappings can run in real time. For outbound mappings, select **Transaction**.
   - Select **All Input** to apply the transformation logic to all incoming data, and to drop the incoming transaction boundaries. Select **All Input** when the results of the transformation depend on all rows of data in the source.
8. Enter the message type and the basic IDoc type, and then click **Fetch IDoc**.

   The segment details of the IDoc appears.
9. On the **IDoc Document** tab, in the **Select IDoc Segments** section, select the segments you want to include in the mapplet.
10. Optionally, in an inbound mapping, to add a control record field to the mapplet, on the **Control Record** tab, select the checkbox for the field you want to add. This enables you to map the selected control record fields when you configure a mapping, integration template, or mapping task.
11. Click **Save**.

Configuring an outbound mapping to read IDocs from SAP

You can configure an outbound mapping to read IDocs from SAP.

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 an SAP 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 select an SAP IDoc Reader connection.

   When you select an SAP IDoc Reader as connection, the source type is a single object and source object is the IDoc Reader Object.

   You can also parameterize the SAP IDoc Reader connection.
6. If required, configure the advanced source properties.

7. To add an IDoc mapplet transformation, on the **Transformation palette**, click Mapplet.
    a. On the **General** tab, enter a name and description for the mapplet.
    b. Draw a link to connect the previous transformation to the transformation.
    c. On the **Mapplet** tab, click **Select**.

        The **Select Mapplet** dialog box appears.
    d. Specify an IDoc mapplet that you imported using the Interpreter transformation and click **OK**.
8. To add any other transformation, on the **Transformation palette**, click the transformation name. Or, drag the transformation onto the mapping canvas.
    a. On the **General** tab, 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 that you create.
    e. To add another transformation, repeat these steps.
9. To add a Target transformation, on the **Transformation palette**, click **Target**.
    a. On the **General** tab, 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 required, configure the advanced target properties.

        Target details and advanced target properties appear based on the connection type.
    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.
10. Save and run the mapping. or save and create a mapping task.

For information about the source and target transformations, see *Transformations*.

Configuring an inbound mapping to write IDocs to SAP

You can configure an inbound mapping to write IDocs to SAP.

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 the source details.
6. To add an IDoc mapplet transformation, on the **Transformation palette**, click Mapplet.
    a. On the **General** tab, enter a name and description for the mapplet.

b. Draw a link to connect the previous transformation to the transformation.

c. On the **Mapplet** tab, click **Select**.

The **Select Mapplet** dialog box appears.

d. Specify an IDoc mapplet that you imported using the Prepare transformation and click **OK**.

e. To preview fields, configure the field rules, or rename fields, click **Incoming Fields**.

Note: You must link the DOCNUM port of the mapplet to the source transformation. The DOCNUM port represents a unique number for each IDoc and the SAP system does not accept inbound IDocs without a unique document number.

7. To add any other transformation, on the **Transformation palette**, click the transformation name. Or, drag the transformation onto the mapping canvas.

a. On the **General** tab, 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 that you create.

e. To add another transformation, repeat these steps.

8. To add a Target transformation, on the **Transformation palette**, click **Target**.

a. On the **General** tab, 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 SAP target details.

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.

9. Save and run the mapping or save and create a mapping task.

Outbound mapping to read MATMAS IDocs from SAP example

You can read material master (MATMAS) IDocs from SAP and write it to a relational database object.

In this example, to read the MATMAS IDocs, perform the following steps:

1. Configure an IDoc mapplet with the MATMAS IDoc.
2. Configure a mapping using the generated mapplet.

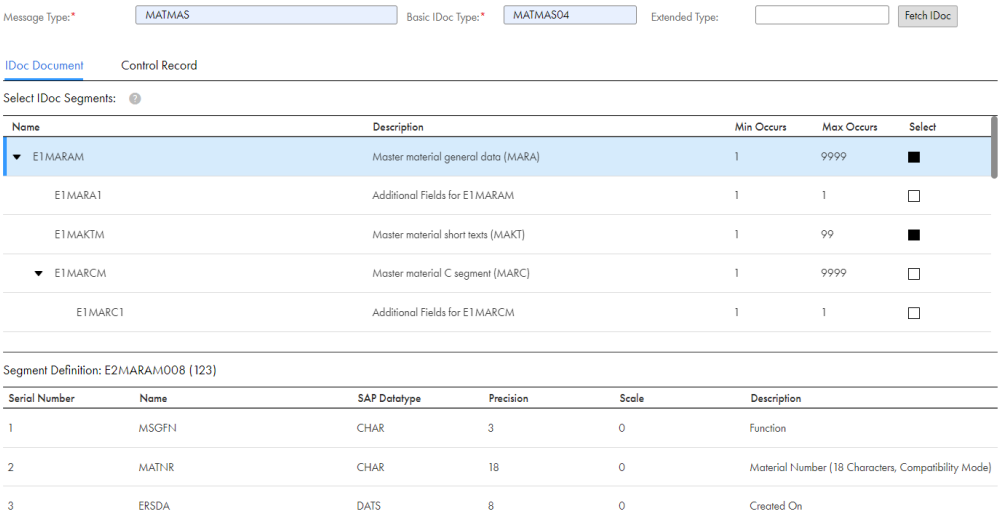
Step 1: Configuring an IDoc mapplet with the MATMAS IDoc

Configure an IDoc mapplet with the MATMAS IDoc from Data Integration. Use the configured mapplet to an outbound mapping.

1. In Data Integration, click **New** > **Mapplets** > **Mapplet - SAP IDoc**, and then click **Create**.
2. Enter a unique name for the mapplet.
3. Specify the location where you save the mapplet.

4. Optionally, enter a description for the mapplet.
5. Select **IDoc Interpreter** as the transformation type.
6. From the **Connection** list, select the source connection by which you fetch the IDoc.
7. From the **Transformation Scope** list, select **Transaction**.
8. In the **Message Type** field, enter MATMAS.
9. In the **Basic IDoc Type** field, enter MATMAS04.
10. Click **Fetch IDoc**.

    The segment details of the IDoc appears in the **Select IDoc Segments** section.



11. On the **IDoc Document** tab, in the **Select IDoc Segments** section, select the segments you want to include in the mapplet.
12. Click **Save**.

Step 2: Configuring an outbound mapping with the MATMAS IDoc

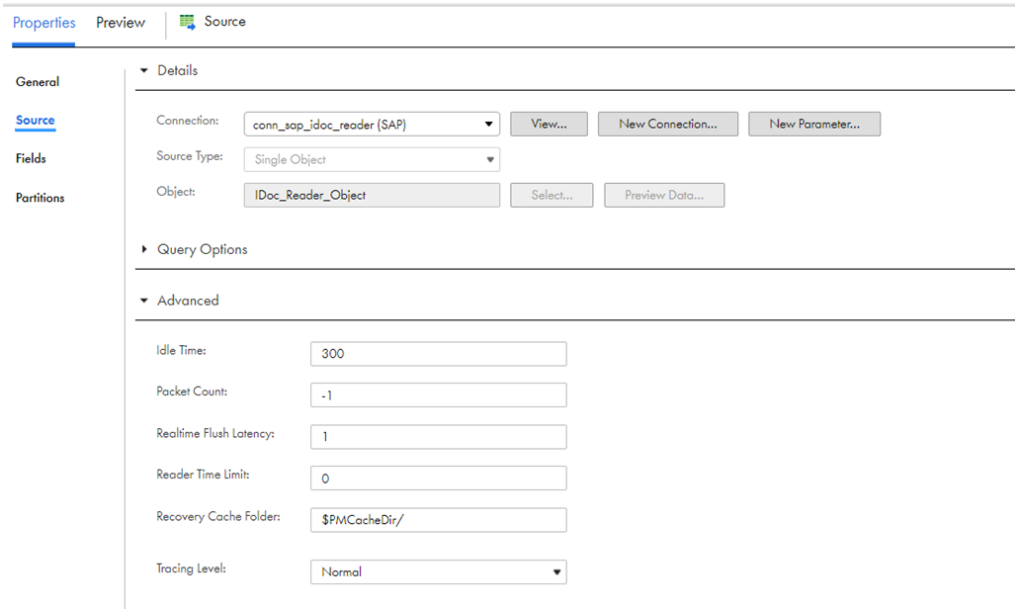
Configure an outbound mapping that uses the IDoc mapplet to read the MATMAS IDoc from SAP.

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**.
3. To configure an SAP 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 select an SAP IDoc Reader connection.

   When you select an SAP IDoc Reader as connection, the source type is a single object and source object is the IDoc Reader Object.

6. If required, configure the advanced source properties.

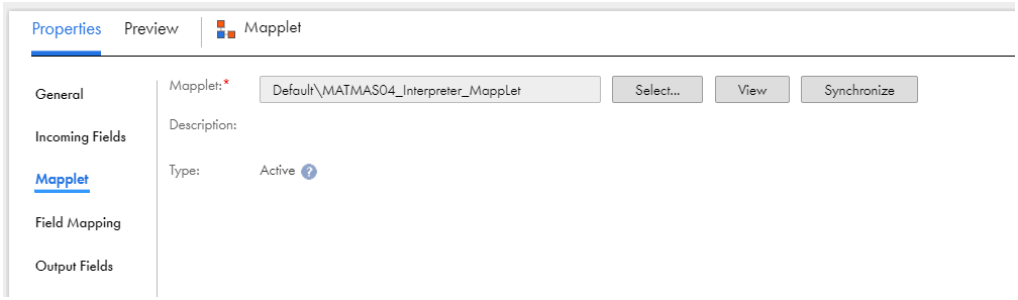
   The following image shows the SAP source details:



7. To add an IDoc mapplet transformation, on the **Transformation palette**, click Mapplet.
8. On the **General** tab, enter a name and description for the mapplet.
9. Draw a link to connect the Source transformation to the Mapplet transformation.
10. On the **Mapplet** tab, click **Select**.

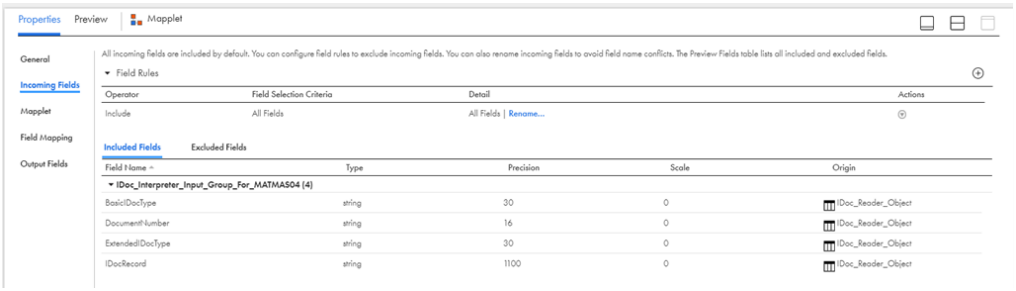
    The **Select Mapplet** dialog box appears.
11. Specify an IDoc mapplet that you imported using the Interpreter transformation and click **OK**.

    The following image shows the MATMAS mapplet details:



12. To preview the incoming single IDoc Reader Object, click **Incoming Fields**.

    The following image shows the incoming single IDoc Reader object:



13. Click **Field Mappings** to map the incoming IDoc Record field with the IDocData field in the mapplet.
14. To add a Target transformation, on the **Transformation palette**, click **Target**.
15. On the **General** tab, enter a name and description.
16. Click the **Target** tab, and configure the required target database details.

    Note: When you use IDoc Listener as a source, you can specify only those targets that honor commit. Contact Global Customer Support for the list of supported targets.
17. Draw a link to connect the mapplet groups to the target database object.
18. Save and run the mapping.

Inbound mapping to write IDocs to SAP example

You can create material master (MATMAS) IDocs in SAP using the MATMAS IDoc type.

In this example, to write MATMAS IDoc to SAP, perform the following steps:

1. Configure an IDoc mapplet with the MATMAS IDoc.
2. Configure a mapping using the generated mapplet.

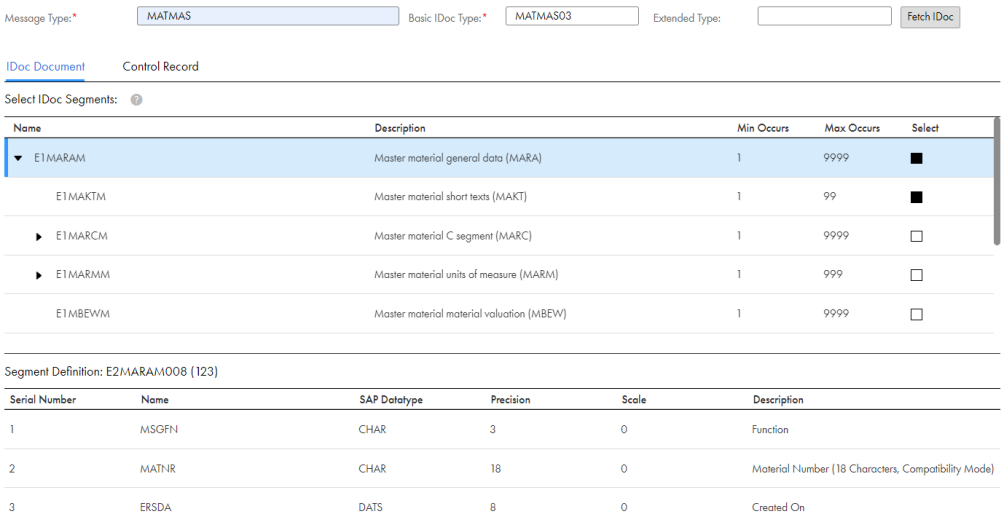
Step 1: Configuring an IDoc mapplet with the MATMAS IDoc

Configure an IDoc mapplet with the MATMAS IDoc from Data Integration. Use the configured mapplet to an inbound mapping.

1. In Data Integration, click **New** > **Mapplets** > **Mapplet - SAP IDoc**, and then click **Create**.
2. Enter a unique name for the mapplet.
3. Specify the location where you save the mapplet.
4. Optionally, enter a description for the mapplet.
5. Select **IDoc Prepare** as the transformation type.

   Note: Do not configure multiple IDoc Prepare transformations for an IDoc mapplet in an inbound mapping on Linux. Otherwise, the mapping fails with the DTM error.
6. From the **Connection** list, select the source connection by which you fetch the IDoc.
7. From the **Transformation Scope** list, select **All Input**.
8. In the **Message Type** field, enter MATMAS.
9. In the **Basic IDoc Type** field, enter MATMAS03.
10. Click **Fetch IDoc**.

The segment details of the IDoc appears in the **Select IDoc Segments** section.



11. On the **IDoc Document** tab, in the **Select IDoc Segments** section, select the segments you want to include in the mapplet.
12. Optionally, to add a control record field to the mapplet, on the **Control Record** tab, select the checkbox for the field you want to add. This enables you to map the selected control record fields when you configure a mapping, integration template, or mapping task.
13. Click **Save**.

Step 2: Configuring an inbound mapping with the MATMAS IDoc

Configure an inbound mapping that uses the IDoc mapplet to write the MATMAS IDoc from SAP.

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**.
3. To configure a flat file 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 select an flat file connection.
6. Add separate flat file sources for the control_input group and segments in the MATMAS IDoc.
7. To add an IDoc Mapplet transformation, on the **Transformation palette**, click Mapplet.
8. On the **General** tab, enter a name and description for the mapplet.
9. On the **Mapplet** tab, click **Select**.

   The **Select Mapplet** dialog box appears.
10. Specify an IDoc mapplet that you imported using the Prepare transformation and click **OK**.
11. Draw a link to connect the Source transformations to the Mapplet transformation. For example, connect the Control_Input source object to the Control_Input_Group for the MATMAS IDoc.

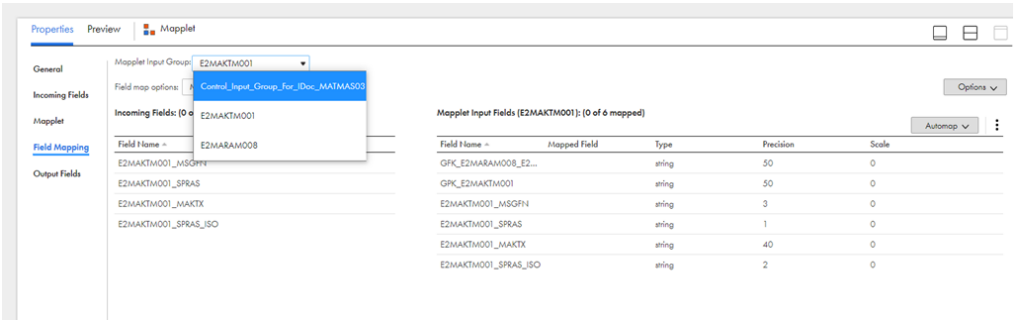
12. To preview the incoming fields, click **Incoming Fields**.

    The following image shows the incoming fields:



13. Click **Field Mappings** to map the incoming fields with the mapplet input fields.

    The following image shows the field mapping in the mapplet:



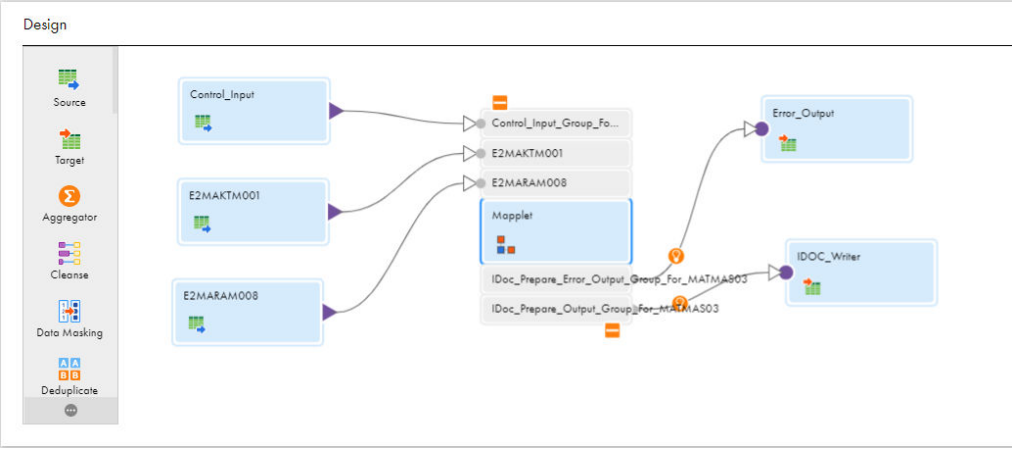
14. To add a Target transformation for writing to SAP, on the **Transformation palette**, click **Target**.
    a. On the **General** tab, enter a name and description.
    b. Click the **Target** tab and configure SAP target details.

    You can also parameterize the SAP target details.

15. To add a flat file Target transformation for the error output, on the **Transformation palette**, click **Target**.
    a. On the **General** tab, enter a name and description.
    b. Click the **Target** tab and configure flat file target details.

16. Draw a link to connect the `IDoc_Prepare_Output_Group_For_MATMAS03` to the IDocWriter object. Draw another link to connect the `IDoc_Prepare_Error_Output_Group_For_MATMAS03` to the Error_Output flat file object.

    The following image shows the mapping for the inbound mapping for the MATMAS IDoc:



17. Save and run the mapping.

SAP mapplet properties

You can configure SAP mapplet properties in an outbound or inbound mapping task when you read IDocs from or write IDocs to SAP.

To configure the SAP mapplet properties, click the **Runtime Options** tab of the outbound or inbound mapping task. Expand the **SAP Mapplet Properties** section, and configure the mapplet properties.

The following table describes the SAP mapplet properties you can configure in an outbound or inbound mapping task:

Property	Outbound / Inbound	Description
Duplicate Parent Row Handling	Outbound and Inbound	Determines how Data Integration handles duplicate parent rows in a mapping task. Select one of the following options: - First Row. Data Integration passes the first duplicate row to the target. Data Integration rejects rows with the same primary key that it processes after this row. - Last Row. Data Integration passes the last duplicate row to the target. - Error. Data Integration passes the first row to the target. Rows that follow with duplicate primary keys increment the error count. The mapping task fails when the error count exceeds the error threshold. Default is First Row.
Orphan Row Handling	Outbound and Inbound	Determines how Data Integration handles orphan rows in a mapping task. Select one of the following options: - Ignore. Data Integration ignores orphan rows. - Error. The mapping task fails when the error count exceeds the error threshold. Default is Ignore.

Property	Outbound / Inbound	Description
Extended Syntax Check	Outbound and Inbound	Checks for IDocs that are not valid. If you use this property to validate IDocs, the performance of the mapping task might degrade.
Row Level Processing	Outbound	Data Integration processes each row of an outbound IDoc according to the IDoc metadata and passes it to a downstream transformation. If you enable row-level processing, Data Integration does not validate IDocs. Select this property to increase performance of the mapping task.
NULL Field Representation	Inbound	Determines how Data Integration handles fields with a null value in the IDoc prepare transformation. Select one of the following options: - Blank. Data Integration inserts all blanks for the field. - Slash (/). Data Integration inserts a single slash (/) for the field.
Cache Directory	Inbound	The directory used to cache inbound IDoc or DMI data. By default, the cache files are created in a directory specified by the variable $PMCacheDir. If you override the directory, make sure the directory exists and contains enough disk space for the cache files.
Cache Size	Inbound	Total memory in bytes allocated to Data Integration to cache data prepared for IDoc prepare transformations. Default is 10 MB.

CHAPTER 5

SAP data type reference

Data Integration uses the following data types in mappings 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 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 IDoc Reader, IDoc Writer, and RFC/BAPI Interface connectors support 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.
CUKY	String	1 to 104,857,600 characters Fixed-length or varying-length string.

SAP Data Type	Transformation Data Type	Range for Transformation Data Type
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 in the following scenarios: - When you configure SAP BAPI Connector as a business service within a mapping or a mapping task to integrate with SAP - When you use SAP BAPI/RFC mapplets to integrate with SAP
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 configure SAP BAPI Connector as a business service within a mapping or a mapping task to integrate with SAP.
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 in the following scenarios: - When you configure SAP BAPI Connector as a business service within a mapping or a mapping task to integrate with SAP - When you use SAP BAPI/RFC mapplets to integrate with SAP
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 in the following scenarios: - When you configure SAP BAPI Connector as a business service within a mapping or a mapping task to integrate with SAP - When you use SAP BAPI/RFC mapplets to integrate with SAP
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.
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.

SAP Data Type	Transformation Data Type	Range for Transformation Data Type
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
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 in the following scenarios: - When you use IDocs to integrate with SAP - When you use SAP BAPI/RFC mapplets to integrate with SAP
SSTRING	String	Small Character string. You can use the SSTRING data type when you use SAP BAPI/RFC mapplets to integrate with SAP.
STRING	String	Character string. You can use the STRING data type in the following scenarios: - When you use SAP BAPI/RFC mapplets to integrate with SAP - When you use IDocs to integrate with SAP
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.

INDEX

B

BAPI/RFC
  configuring mappings 27
  map error output 25
  mapping example 28
  nested structure 23
BAPI/RFC mappings
  configuring 27
  configuring mapping example 30
  example 28
  generate mapplet example 29
  overview 25
BAPI/RFC mapplets
  integration ID 24
  parameters 22
  rules and guidelines 25
BAPI/RFC parameter
  properties 22
  system variables 24

C

Cloud Application Integration community
  URL 5
Cloud Developer community
  URL 5
connections
  SAP IDoc Reader 17
  SAP IDoc Writer 18
  SAP RFC/BAPI Interface 19

D

Data Integration community
  URL 5
data types
  SAP 55

I

IDoc
  generating metadata file 37
IDoc mappings
  overview 42
  read example 47, 50
IDoc mapplets
  overview 35
IDoc Reader mappings
  sources 42
IDoc Writer mappings
  targets 44
IDocs
  configuring mapplet 47
  importing metadata 41, 44
  mapping read example 47, 50
  MATMAS IDoc 50
  primary and foreign keys 39
  properties 36
inbound mappings
  configuring to write IDocs 46
  configuring to write IDocs example 51
Informatica Global Customer Support
  contact information 6
Informatica Intelligent Cloud Services
  web site 5

M

maintenance outages 6
mappings
  IDoc Reader sources 42
  IDoc Writer targets 44
mapplet
  outbound 38
mapplets
  BAPI/RFC 21
  inbound 39
metadata
  generating file for IDoc 37

O

outbound mappings
  configuring to read IDocs 45
  configuring to read IDocs example 48
outbound mapplet
  map error output 38
  ports 38

R

RSEIDoc3
  generating metadata file for IDoc 37
rules and guidelines
  BAPI/RFC mapplets 25

S

SAP
  data types 55
SAP connections
  IDoc and BAPI/RFC 17

SAP Connector
assets 8
communication interfaces 8
integration methods 7
overview 7
transformations 8
SAP IDoc Reader
connection properties 17
SAP IDoc Writer
connection properties 18
SAP integration methods
overview 7
using BAPI/RFC functions 21
using IDocs 34
SAP mapplet properties
BAPI/RFC mapping task 32
outbound or inbound mapping task 53
SAP RFC/BAPI Interface
connection properties 19
segments and groups
overview 35
status
Informatica Intelligent Cloud Services 6
system status 6

T

trust site
description 6

U

upgrade notifications 6

W

web site 5