Informatica®

Informatica® Cloud Data Integration
July 2024

Mappings

Informatica Cloud Data Integration Mappings
July 2024

© Copyright Informatica LLC 2006, 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, Informatica Cloud, Informatica Intelligent Cloud Services, PowerCenter, PowerExchange, and the Informatica logo 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.

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: 2024-07-12

Table of Contents

Preface 7

Informatica Resources 7

- Informatica Documentation 7
- Informatica Intelligent Cloud Services web site 7
- Informatica Intelligent Cloud Services Communities 7
- Informatica Intelligent Cloud Services Marketplace 7
- Data Integration connector documentation 8
- Informatica Knowledge Base 8
- Informatica Intelligent Cloud Services Trust Center 8
- Informatica Global Customer Support 8

Chapter 1: Mappings 9

Mapping Designer 10

Mapping templates 14

- Industry solutions templates 16

Mapping configuration 17

- Defining a mapping 18
- Configuring the source 18
- Configuring the data flow 19
- Configuring the target 20
- Rules and guidelines for mapping configuration 20
- Rules and guidelines for mappings on GPU-enabled clusters 21
- Data flow run order 21

Mappings in advanced mode 23

- Using an advanced cluster 24
- Mapping configuration in advanced mode 24
- Runtime plans 26

Mappings in SQL ELT mode 26

- Sources in mappings in SQL ELT mode 27
- Targets in mappings in SQL ELT mode 28
- Transformations in mappings in SQL ELT mode 28
- Mapping behavior differences 29
- Mapping configuration in SQL ELT mode 29

Automapping 30

- Generate automapping 31
- Transformations in automapping 31

Mapping validation 31

- Validating a mapping 32
- Validating a mapping in advanced mode 33
- Validating a mapping in SQL ELT mode 33

Data preview in mappings. . . . 34
Preview behavior for a mapping. . . . 35
Running a preview job for a mapping. . . . 35
Viewing preview results for a mapping. . . . 36
Customizing preview results. . . . 37
Data preview in advanced mode. . . . 37
Running a preview job in advanced mode. . . . 38
Viewing preview results in advanced mode. . . . 39
Detailed data preview in mapplets. . . . 39
Running detailed data preview in a mapplet. . . . 40
SQL ELT query preview. . . . 41
Running an SQL ELT query preview for mappings configured for SQL optimization. . . . 42
Running an SQL ELT query preview for mappings in SQL ELT mode. . . . 42
SQL ELT query preview results files. . . . 43
SQL ELT query data preview. . . . 43
Running an SQL ELT query data preview job. . . . 44
Field lineage. . . . 44
Lineage for renamed fields. . . . 45
Lineage for fields containing expressions. . . . 45
Lineage for mapped fields. . . . 46
Mapplet field lineage. . . . 46
Lineage for lookup fields. . . . 47
Lineage for transformations that read Data Quality assets. . . . 47
Viewing field lineage. . . . 48
Testing a mapping. . . . 48
Mapping maintenance. . . . 49
Mapping revisions and mapping tasks. . . . 49
Bigint data conversion. . . . 50

Chapter 2: Mapping tutorial. . . . 51

Preparing for the mapping tutorial. . . . 52
Step 1. Create a mapping. . . . 52
Step 2. Configure a source. . . . 53
Step 3. Create a Filter transformation. . . . 54
Step 4. Configure a target. . . . 56
Step 5. Validate and test the mapping. . . . 57
Step 6. Create a mapping task. . . . 60

Chapter 3: Parameters. . . . 62

Input parameters. . . . 62
Input parameter types. . . . 65
Input parameter configuration. . . . 67
Partial parameterization with input parameters. . . . 68

Using parameters in a mapping. . . . 69
In-out parameters. . . . 70
Aggregation types. . . . 72
Variable functions. . . . 72
In-out parameter properties. . . . 73
In-out parameter values. . . . 74
Rules and guidelines for in-out parameters. . . . 75
Creating an in-out parameter. . . . 75
Editing in-out parameters in a mapping task. . . . 76
In-out parameter example. . . . 78
In-out parameter example in advanced mode. . . . 80
Using in-out parameters as expression variables. . . . 83
Using in-out parameters as source queries. . . . 83
Parameter files. . . . 84
Parameter file requirements. . . . 85
Parameter scope. . . . 86
Sample parameter file. . . . 87
Parameter file location. . . . 87
Rules and guidelines for parameter files. . . . 90
Parameter file templates. . . . 91
Overriding connections with parameter files. . . . 91
Overriding data objects with parameter files. . . . 92
Overriding source queries. . . . 93
Creating target objects at run time with parameter files. . . . 93

Chapter 4: CLAIRE recommendations. . . . 95

Transformation type recommendations. . . . 96
Source recommendations. . . . 96
Join recommendations. . . . 97
Union recommendations. . . . 98
Mapping inventory recommendations. . . . 98
Component recommendations. . . . 98
Mapplet recommendations. . . . 98
User-defined function recommendations. . . . 99
Data standardization recommendations. . . . 99
Replace NULL values. . . . 99
Date type standardization. . . . 100
Data masking recommendations. . . . 100
Viewing data masking recommendations. . . . 101
Rule specification recommendations. . . . 101

Chapter 5: Data catalog discovery. . . . 102

Performing data catalog discovery. . . . 103

Mapping inventory. 104
Catalog search. 105
Discovering and selecting a catalog object. 106
Data catalog discovery example. 107

Chapter 6: Visio templates. 108

Visio template revisions. 108
Configuring parameter properties. 108
Downloading a template XML file. 112
Deleting a Visio template. 112

Index. 113

Preface

Use *Mappings* to learn how to create and use mappings in Informatica Cloud® Data Integration to define the flow of data from sources to targets. *Mappings* also contains information about finding and using objects in your data catalog and information about creating and using parameters.

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

Mappings

A mapping defines reusable data flow logic that you can use in mapping tasks. Use a mapping to define data flow logic that is not available in synchronization tasks, such as specific ordering of logic or joining sources from different systems.

You can create the following types of mappings:

Mapping

Create a mapping when you need flexibility in your sources, targets, and transformation options. A mapping can read and write to a wide variety of heterogeneous data sources. It also offers a large variety of data transformation options.

When you run a mapping, Data Integration processes some or all of the transformation logic. You can choose to push some or all the transformation logic to the source, to the target, or both. Data Integration processes any transformation logic that isn't pushed to the sources and targets.

Mapping in advanced mode

Create a mapping in advanced mode when you want to process multilevel hierarchical data, embedded code snippets, and workloads at any scale.

A mapping in advanced mode requires an advanced cluster to run the mapping logic. When you start running mappings in advanced mode, Data Integration can automatically create a local advanced cluster for you to use.

When you run a mapping in advanced mode, you can choose to push some or all the transformation logic to the source, to the target, or both. Data Integration processes any transformation logic that isn't pushed to the sources and targets.

Mapping in SQL ELT mode

Create a mapping in SQL ELT mode when your source and target are in the same cloud ecosystem and you want to perform the data transformation entirely within the cloud ecosystem. For example, you want to read data from your Snowflake cloud data warehouse or data lake, load it to your Snowflake cloud data warehouse, and perform all of the data transformation within Snowflake.

When you run a mapping in SQL ELT mode, Data Integration translates the transformation logic into ecosystem-specific SQL statements and commands that run in the underlying cloud data warehouse. This increases the data processing speed because the data isn't moved out of the cloud infrastructure for processing. It also increases the efficiency of the data integration pipelines.

Use the Mapping Designer to configure mappings. When you configure a mapping, you describe the flow of data from source to target. You can add transformations to transform data, such as an Expression transformation for row-level calculations or a Filter transformation to remove data from the data flow. A transformation includes field rules to define incoming fields. Links visually represent how data moves through the data flow.

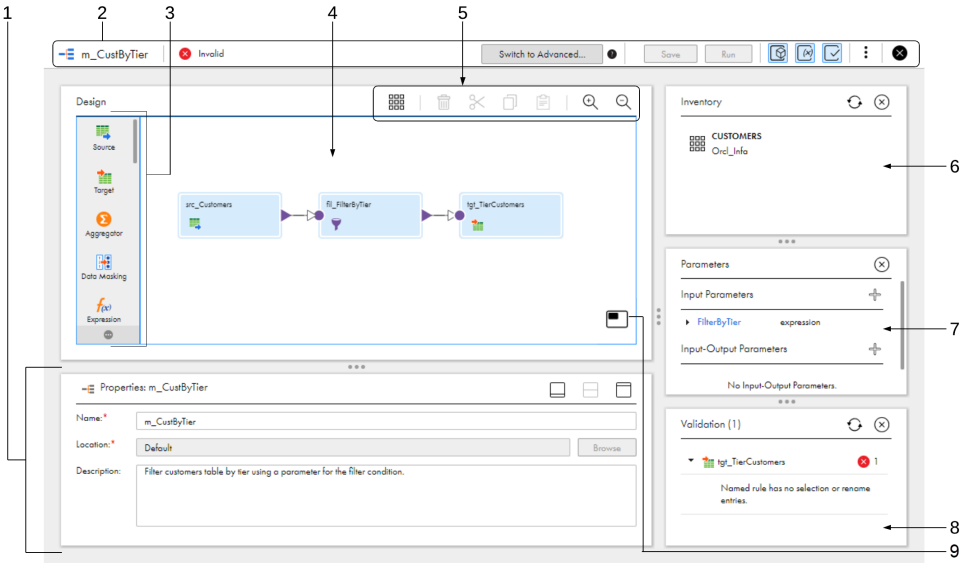
You can configure parameters to enable additional flexibility in how you can use the mapping. Parameters act as placeholders for information that you define in the mapping task. For example, you can use a parameter for a source connection in a mapping, and then define the source connection when you configure the task.

You can use components such as mapplets, business services, and hierarchical schema definitions in mappings. Components are assets that support mappings. Some components are required for certain transformations while others are optional. For example, a business services asset is required for a mapping that includes a Web Service transformation. Conversely, a saved query component is useful when you want to reuse a custom query in multiple mappings, but a saved query is not required.

Mapping Designer

Use the Mapping Designer to create mappings that you can use in mapping tasks.

The following image shows the Mapping Designer:

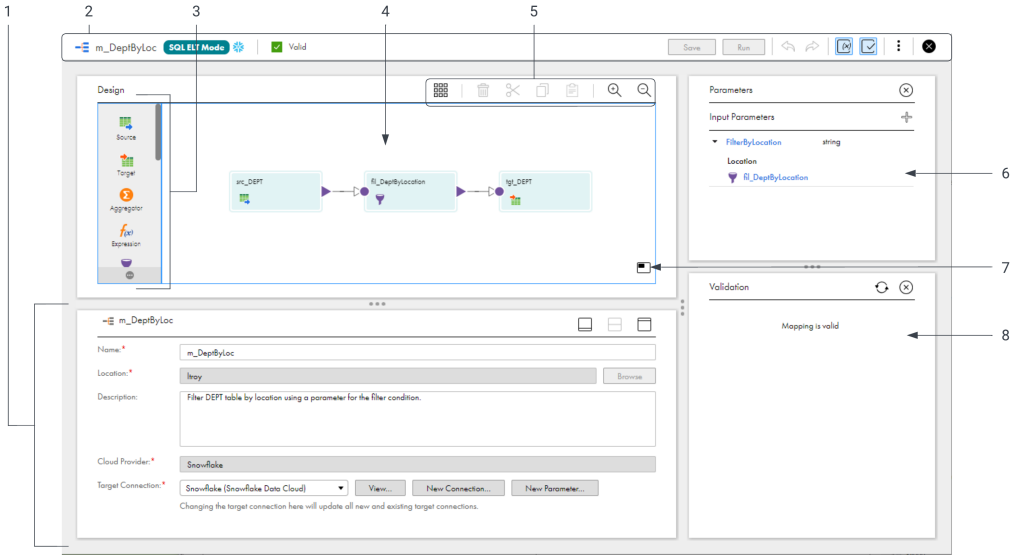


The following table describes the areas of the Mapping Designer:

Mapping Designer Areas	Description
1. Properties panel	Displays configuration options for the mapping or selected transformation. Different options display based on the transformation type. Includes icons to quickly resize the **Properties** panel. Use the icons to display the **Properties** panel, the mapping canvas, or both. You can also manually resize the **Properties** panel.
2. Header	Displays the following information, buttons, and icons: - Mapping name. - Mapping status. The status can be either Valid or Invalid. - Switch to Advanced button. Use to update the mapping canvas to advanced mode. - Save button. - Run button. Use to create a test run of the mapping. - Undo and redo icons. You can undo and redo actions until you save the mapping. - Inventory icon. Displays and hides the **Inventory** panel. - Parameters icon. Displays and hides the **Parameters** panel. - Validation icon. Displays and hides the **Validation** panel. - SQL ELT Query icon. Displays and hides the **SQL ELT Query** panel. - Actions icon. Use to create and save a mapping task based on the mapping. See *Tasks* for more information about mapping tasks. - Close icon.
3. Transformation palette	Lists the transformations that you can use in the mapping. To add a transformation, drag the transformation to the mapping canvas.
4. Mapping canvas	The canvas where you configure a mapping. When you create a mapping, a Source transformation and a Target transformation are already on the canvas for you to configure.
5. Toolbar	Displays the following icons and buttons: - Recommendation toggle. Turns CLAIRE™ recommendations on or off. Shown when organization CLAIRE recommendation preferences are enabled. - Arrange All icon. Arranges the mapping. - Delete icon. Deletes the selected transformation or link. - Cut. - Copy. - Paste. - Zoom In icon. Increases the size of the mapping. - Zoom Out icon. Decreases the size of mapping.
6. Inventory panel	Lists Enterprise Data Catalog objects that you can add to the mapping as sources, targets, or lookup objects. **Note:** Your organization must have the appropriate licenses in order for you to see the **Inventory** panel. Objects appear in the inventory when you search for objects on the **Data Catalog** page, select them in the search results, and add them to the mapping. To delete an object from the inventory, click "X" in the row that contains the object. Displays when you click **Inventory**. To hide the panel, click **Inventory** again.
7. Parameters panel	Lists the parameters in the mapping. You can create, edit, and delete parameters, and see where the mapping uses parameters. Displays when you click **Parameters**. To hide the panel, click **Parameters** again.

Mapping Designer Areas	Description
8. Validation panel	Lists the transformations in the mapping and displays details about mapping errors. Use to find and correct mapping errors. Displays when you click **Validation**. To hide the panel, click **Validation** again.
9. Bird's Eye View icon	Opens the **Bird's Eye View** panel which displays an overview of your entire mapping and highlighted field lineage. Use the navigation pane in the panel to scroll the canvas.

For mappings in SQL ELT mode, the Mapping Designer looks slightly different, as shown in the following image:



The following table describes the areas of the Mapping Designer for a mapping in SQL ELT mode:

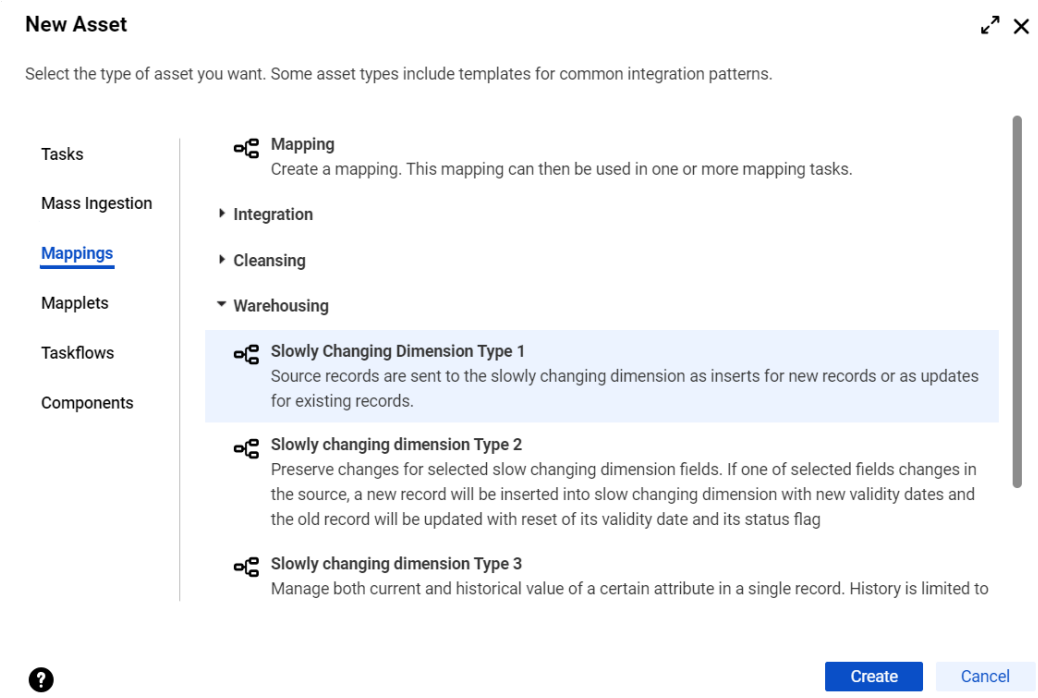
Mapping Designer Areas	Description
1. Properties panel	Displays configuration options for the mapping or selected transformation. Different options display based on the transformation type. Includes icons to quickly resize the **Properties** panel. Use the icons to display the **Properties** panel, the mapping canvas, or both. You can also manually resize the **Properties** panel.
2. Header	Displays the following information, buttons, and icons: - Mapping name. - "SQL ELT Mode" indicator and icon for the cloud ecosystem. - Mapping status. The status can be either Valid or Invalid. - Save button. - Run button. In SQL ELT mode, the Run button is disabled. To run the mapping, create a mapping task. - Undo and redo icons. You can undo and redo actions until you save the mapping. - Parameters icon. Displays and hides the **Parameters** panel. - Validation icon. Displays and hides the **Validation** panel. - Actions icon. Use to create and save a mapping task based on the mapping. See *Tasks* for more information about mapping tasks. - Close icon.
3. Transformation palette	Lists the transformations that you can use in the mapping. To add a transformation, drag the transformation to the mapping canvas.
4. Mapping canvas	The canvas where you configure a mapping. When you create a mapping, a Source transformation and a Target transformation are already on the canvas for you to configure.
5. Toolbar	Displays the following icons and buttons: - Arrange All icon. Arranges the mapping. - Delete icon. Deletes the selected transformation or link. - Cut. - Copy. - Paste. - Zoom In icon. Increases the size of the mapping. - Zoom Out icon. Decreases the size of mapping.
6. Parameters panel	Lists the parameters in the mapping. You can create, edit, and delete parameters, and see where the mapping uses parameters. Displays when you click **Parameters**. To hide the panel, click **Parameters** again.
7. Bird's Eye View icon	Opens the **Bird's Eye View** panel which displays an overview of your entire mapping and highlighted field lineage. Use the navigation pane in the panel to scroll the canvas.
8. Validation panel	Lists the transformations in the mapping and displays details about mapping errors. Use to find and correct mapping errors. Displays when you click **Validation**. To hide the panel, click **Validation** again.

Mapping templates

You can use a mapping template instead of creating a mapping from scratch.

Mapping templates are divided into the categories: Integration, Cleansing, Warehousing, and Industry Solutions.

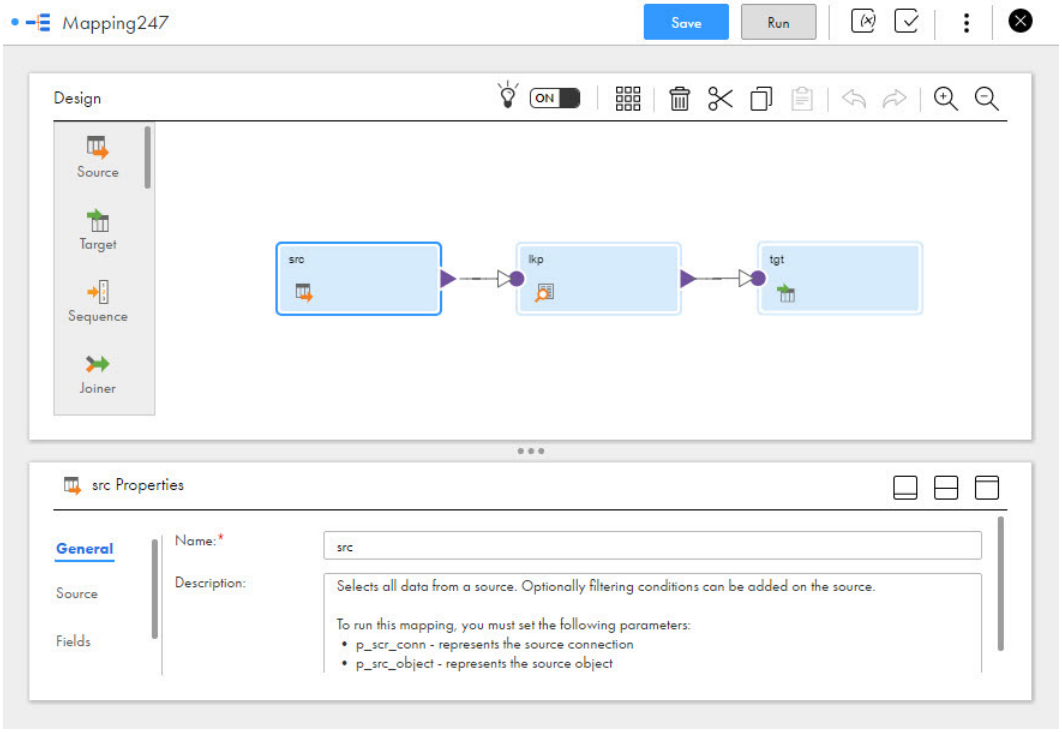
The following image shows some of the mapping templates in the New Asset dialog box:



When you select a mapping template in the **New Asset** dialog box, you create a mapping that uses a copy of the mapping template.

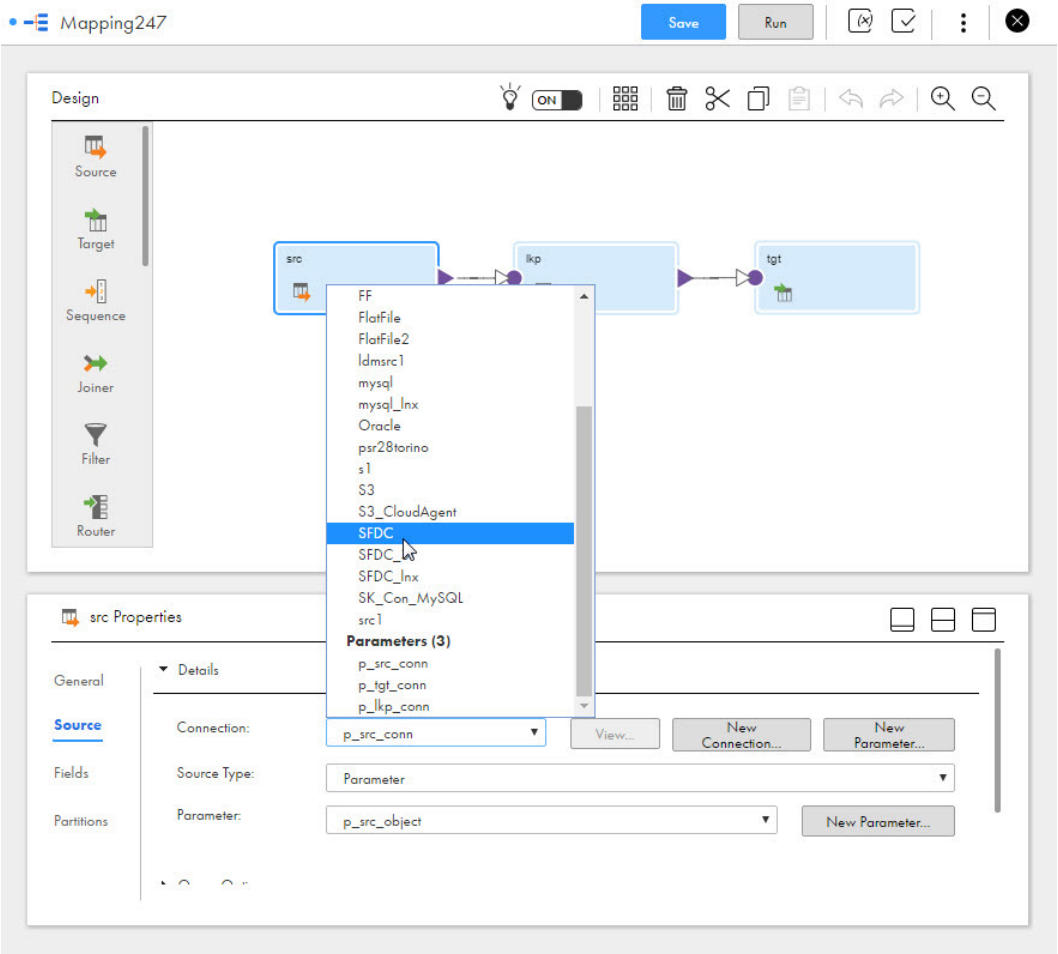
The template mapping contains pre-populated transformations. Click on each of the transformations in the mapping to see the purpose of the transformation, how the transformation is configured, and which parameters are used.

The following image shows the Augment data with Lookup template with the Source transformation selected. The **Description** field shows how the Source transformation is configured:



You can use a mapping template as is or you can reconfigure the mapping. For example, the Augment data with Lookup template uses the p_scr_conn parameter for the source connection. You can use the parameter to specify a different connection each time you run the mapping task that uses this mapping. You might want

to use the same source connection every time you run the mapping task. You can replace the parameter p_scr_conn with a specific connection, as shown in the following image:



When you save the mapping, you save a copy of the template. You do not modify the template itself.

You can copy a template mapping to advanced mode. Mapping templates don't apply to mappings in SQL ELT mode.

Industry solutions templates

Use an industry solutions mapping template to create a mapping that processes incoming EDI, HL7, or HIPAA messages. The templates are pre-configured to use industry solutions data services to process your incoming messages.

The Industry Solutions category includes mapping templates for the following scenarios:

- Process incoming EDI X12 messages
- Process incoming HL7 messages
- Process incoming HL7 messages from a Kafka source
- Process incoming HL7 messages to a FHIR bundle
- Process incoming HIPAA X12 messages

Each template includes a pre-configured Expression transformation that extracts fields from your incoming messages and uses them to build the data service name according to the message type and version. It then

passes the names to the Data Services transformation. The Expression transformation allows you to use a single mapping to process all of your incoming EDI, HL7, or HIPAA messages.

The Data Services transformation invokes data services from the data services repository. The data services parse the incoming messages and output an XML structure that represents the parsed messages.

Process incoming HL7 messages to a FHIR bundle

Use a mapping template instead of creating a mapping from scratch to process incoming HL7 messages and transform them into a FHIR-R4 bundle in XML format.

The template includes a Data Services transformation that invokes the HL7-FHIR mapper service. The mapper service maps HL7 message components to corresponding FHIR message components. The mapping between components is based on HL7 v2.6, and FHIR version 4.0.1. The resulting bundle is a collection of related FHIR resources. The template does not include the FHIR provenance resource.

Note: The FHIR standard is continuously developing. This template is based on the HL7 V2 to FHIR mapping implementation available as of the April 2023 release of Data Integration. Messages, segments, fields, data types, or code sets that you use may not be available. As new versions of FHIR are released, this template might be updated to reflect additional HL7 to FHIR mappings.

For more information about the standard, see the
HL7 V2 to FHIR Implementation Guide on the HL7 International website.

The HL7 to FHIR mapping template is configured to process the following HL7 events:

- ADT_A01
- ADT_A04
- ADT_A08
- OML_O21
- ORM_O01
- VXU_V04
- ORU_R01

Mapping configuration

Use the Mapping Designer to configure a mapping.

To configure a mapping, perform the following tasks:

1. Define the mapping.
2. Configure the source.
3. Configure the data flow. Add, and configure transformations, and draw links to represent the flow of data.

   Tip: You can copy and paste transformations within or between open mappings.
4. Configure the target.
5. Optionally, create parameters to be defined in the mapping task.
6. Save and validate the mapping.

Defining a mapping

To define a mapping, create or edit a mapping, specify the mapping name and location, and optionally enter a mapping description.

1. Click **New** > **Mappings**, and then perform one of the following tasks:
   - To create a mapping from scratch, click **Mapping** and then click **Create**. In the **New Mapping** dialog box, select the mapping type and then click **Continue**.

     If you create a mapping in SQL ELT mode, select the target connection or create a new target connection and then click **Next**.
   - To create a mapping based on a template, click the template you want to use and then click **Create**.
   - To edit a mapping, on the **Explore** page, navigate to the mapping. In the row that contains the mapping, click **Actions** and select **Edit**.

   The Mapping Designer opens.
2. To specify the mapping name and location, in the **Mapping Properties** panel, enter a name for the mapping and change the location. Or, you can use the default values if desired.

   The default mapping name is *Mapping* followed by a sequential number.

   Mapping names can contain alphanumeric characters and underscores (_). Maximum length is 100 characters.

   The following reserved words cannot be used:
   - AND
   - OR
   - NOT
   - PROC_RESULT
   - SPOUTPUT
   - NULL
   - TRUE
   - FALSE
   - DD_INSERT
   - DD_UPDATE
   - DD_DELETE
   - DD_REJECT

   If the **Explore** page is currently active and a project or folder is selected, the default location for the asset is the selected project or folder. Otherwise, the default location is the location of the most recently saved asset.

   You can change the name or location after you save the mapping using the **Explore** page.
3. Optionally, enter a description of the mapping.

   Maximum length is 4000 characters.

Configuring the source

To configure the source, edit the Source transformation.

1. On the mapping canvas, click the Source transformation.
2. To specify a name and description for the Source transformation, in the **Properties** panel, click **General**.

Transformation names can contain alphanumeric characters and underscores (_). Maximum length is 75 characters.

The following reserved words cannot be used:

- AND
- OR
- NOT
- PROC_RESULT
- SPOUTPUT
- NULL
- TRUE
- FALSE
- DD_INSERT
- DD_UPDATE
- DD_DELETE
- DD_REJECT

You can enter a description if desired.

Maximum length is 4000 characters.

3. Click the **Source** tab and configure source details, query options, and advanced properties.

   Source details, query options, and advanced properties vary based on the connection type. For more information, see the help for the appropriate connector.

   In the source details, select the source connection and source object. For some connection types, you can select multiple source objects.

   If your organization administrator has configured Enterprise Data Catalog integration properties and you've added objects to the mapping from the **Data Catalog** page, you can select the source object from the **Inventory** panel. The **Inventory** panel isn't available in mappings in SQL ELT mode.

   You can also configure parameters for the source connection and source object.

   In advanced mode, you can add an intelligent structure model to the Source transformation for some source types that use an Amazon S3 or Microsoft Azure Data Lake connection. You must create the model before you can add it to the mapping. For more information about intelligent structure models, see *Components*.

4. To configure a source filter or sort options, expand **Query Options**. Click **Configure** to configure a filter or sort option.
5. Click the **Fields** tab to add or remove source fields, to update field metadata, or to synchronize fields with the source.

   You can't update field metadata for mappings in SQL ELT mode.

6. To save your changes and continue, click **Save**.

Configuring the data flow

To configure the data flow, optionally add transformations to the mapping.

1. To add a transformation, perform either of the following actions:

- On the transformation palette, drag the transformation onto the mapping canvas. If you drop the transformation between two transformations that are connected, the Mapping Designer automatically connects the new transformation to the two transformations.
- On the mapping canvas, hover over the link between transformations or select an unconnected transformation and click the Add Transformation icon. Then select a transformation from the menu.

2. On the **General** tab, enter a name and description for the transformation.
3. Link the new transformation to appropriate transformations on the canvas.

   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.
4. To preview fields, configure the field rules, or rename fields, click **Incoming Fields**.
5. Configure additional transformation properties, as needed.

   The properties that you configure vary based on the type of transformation you create. For more information about transformations and transformation properties, see *Transformations*.
6. To save your changes and continue, click **Save**.
7. To add another transformation, repeat these steps.

Configuring the target

To configure the target, edit the Target transformation.

1. On the mapping canvas, click the **Target** transformation.
2. Link the Target transformation to the appropriate upstream transformation.
3. On the **General** tab, enter the target name and optional description.
4. Click the **Incoming Fields** tab to preview incoming fields, configure field rules, or rename fields.
5. Click the **Target** tab and configure target details and advanced properties.

   Target details and advanced target properties vary based on the connection type. For more information, see the help for the appropriate connector.

   In the target details, select the target connection, target object, and target operation. In mappings in SQL ELT mode, the target connection is automatically configured with the connection you selected when you created the mapping.

   If your organization administrator has configured Enterprise Data Catalog integration properties, and you have added objects to the mapping from the **Data Catalog** page, you can select the target object from the **Inventory** panel. The **Inventory** panel isn't available in mappings in SQL ELT mode.

   You can also configure parameters for the target connection and target object.
6. Click **Field Mapping** and map the fields that you want to write to the target.

Rules and guidelines for mapping configuration

Use the following rules and guidelines when you configure a mapping:

- A mapping does not need a Source transformation if it includes a Mapplet transformation and the mapplet includes a source.
- If it's not a mapping in SQL ELT mode, you can configure multiple branches within the data flow. If you create more than one data flow, configure the flow run order.
- Connect all transformations to the data flow.

- You can merge multiple upstream branches through a passive transformation only when all transformations in the branches are passive.
- When you rename fields, update conditions and expressions that use the fields. Conditions and expressions, such as a Lookup condition or expression in an Expression transformation, do not inherit field name changes.
- To use a connection parameter and a specific object, use a connection and object in the mapping. When the mapping is complete, you can replace the connection with a parameter.
- When you use a parameter for an object, use parameters for all conditions or field mappings in the data flow that use fields from the object.
- You can copy and paste multiple transformations at once between the following open assets:
  - Between mappings
  - Between mapplets
  - From a mapping to a mapplet

  When you paste a transformation into another asset, all transformation attributes except parameter values are copied to the asset.

Rules and guidelines for mappings on GPU-enabled clusters

Use the following rules and guidelines when you configure a mapping that runs on a GPU-enabled advanced cluster:

- The size of a partition file must be smaller than the GPU memory size. To check the GPU memory size, refer to the AWS documentation for the selected worker instance type.
- The mapping cannot read from an Amazon Redshift source or a source based on an intelligent structure model.
- The mapping cannot write to a CSV file.
- If the mapping includes NaN values, the output is unpredictable.
- The mapping cannot process timestamp data types from a Parquet source.
- If you need to process decimal data from a CSV file, read the data as a string and then convert it to a float.
- When the mapping uses an Expression transformation, you can use only scientific functions and the following numeric functions:
  - ABS
  - CEIL
  - FLOOR
  - LN
  - LOG
  - POWER
  - SQRT

To learn how to configure a GPU-enabled cluster, see the Administrator help.

Data flow run order

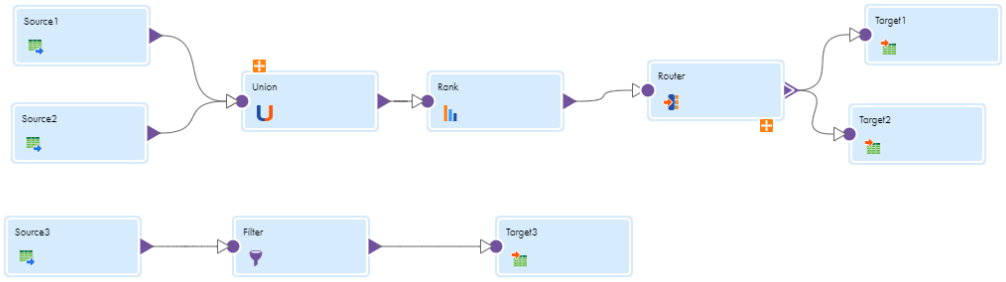
You can specify the order in which Data Integration runs the individual data flows in a mapping. Specify the flow run order when you want Data Integration to load the targets in the mapping in a particular order. For

example, you might want to specify the flow run order when inserting, deleting, or updating tables with primary or foreign key constraints.

You might want to specify the flow run order to maintain referential integrity when updating tables that have primary or foreign key constraints. Or, you might want to specify the flow run order when you are processing staged data.

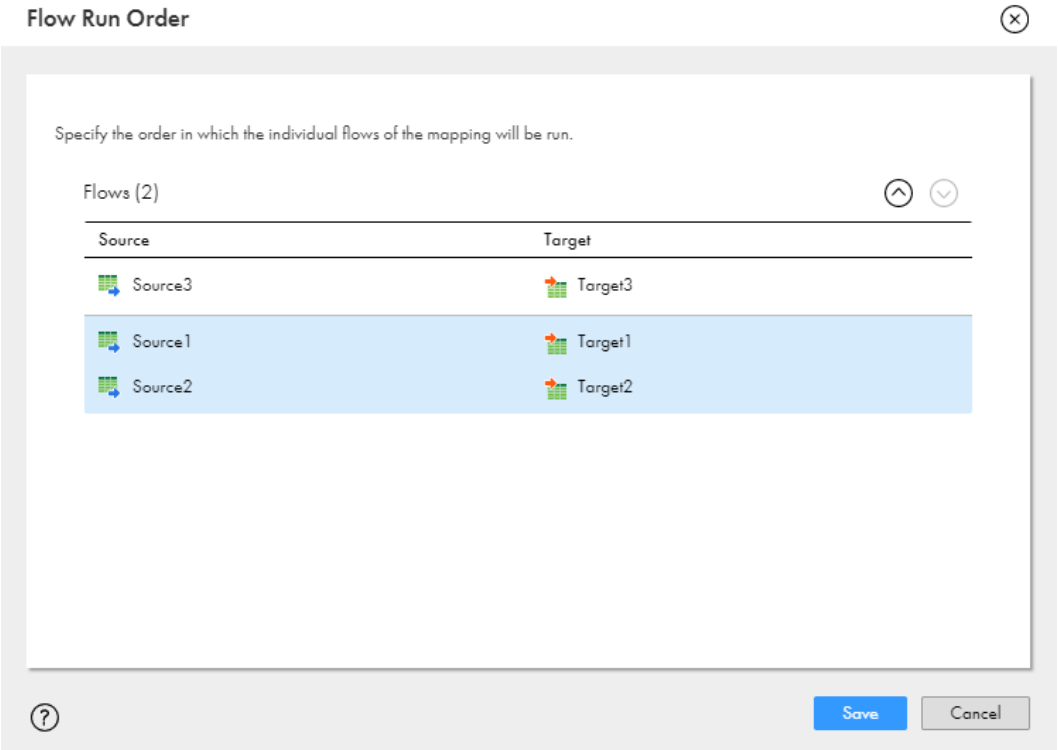
If a flow contains multiple targets, you cannot configure the load order of the targets within the flow.

The following image shows a mapping with two data flows:



In this example, the top flow contains two pipelines and the bottom flow contains one pipeline. A pipeline is a source and all the transformations and targets that receive data from that source. When you configure the flow run order, you cannot configure the run order of the pipelines within a data flow.

The following image shows the flow run order for the mapping:



In this example, Data Integration runs the top flow first, and loads Target3 before running the second flow. When Data Integration runs the second flow, it loads Target1 and Target2 concurrently.

If you add another data flow to the mapping after you configure the flow run order, the new flow is added to the end of the flow run order by default.

If the mapping contains a mapplet, Data Integration uses the data flows in the last version of the mapplet that was synchronized. If you synchronize a mapplet and the new version adds a data flow to the mapping, the new flow is added to the end of the flow run order by default. You cannot specify the flow run order in mapplets.

You can specify the flow run order for data flows with any target type. You can define the data flow run order during SQL ELT optimization in a mapping that uses the following connections:

- Amazon Redshift V2
- Databricks Delta
- Microsoft Azure Synapse SQL
- ODBC
- Snowflake Data Cloud

Note: You can also specify the run order of data flows in separate mapping tasks with taskflows. Configure the taskflow to run the tasks in a specific order. For more information about taskflows, see *Taskflows*.

You can't specify the data flow run order in mappings in advanced mode or SQL ELT mode because these mappings can't have multiple flows.

Configuring the data flow run order

Configure the order in which Data Integration runs the data flows in a mapping.

1. In the Mapping Designer, click **Actions** and select **Flow Run Order**.
2. In the **Flow Run Order** dialog box, select a data flow and use the arrows to move it up or down.
3. Click **Save**.

Mappings in advanced mode

Create a mapping in advanced mode when you want to process multilevel hierarchical data, embedded code snippets, and workloads at any scale. The Mapping Designer updates the mapping canvas to include the transformations and functions that enable advanced functionality.

A mapping in advanced mode can perform the following types of complex processing:

- Read hierarchical or relational input and convert it to relational, hierarchical, or flattened denormalized output using the Hierarchy Processor transformation.
- Embed code snippets in a mapping using the Python and Java transformations.
- Pass data to a machine learning model using the Machine Learning transformation.
- Reprocess incrementally-loaded source files to create a snapshot of the data from a specified time interval, to debug and discover the source of bad data in your target, and to restore deleted data.
- Use a code task API to run hand-coded Scala jobs.
- Use native NoSQL connectors, such as MongoDB, DynamoDB, and CosmosDB, to connect to databases without using third-party drivers.
- Process data using any connector.

A mapping in advanced mode requires an advanced cluster to run mapping logic. When you start running mappings in advanced mode, Data Integration can automatically create a local advanced cluster for you to use.

Note: To enable local clusters, contact your system administrator to update the Secure Agent configuration. For more information, see Prepare for local clusters in the Administrator help.

Because an advanced cluster is required, the mapping must run in one of the following types of runtime environments:

- A Secure Agent group with one Secure Agent. The Secure Agent must be installed on Linux.
- A serverless runtime environment.

Using an advanced cluster

Mappings in advanced mode run on an advanced cluster, which is a Kubernetes cluster that runs mapping logic in a distributed processing environment.

Advanced clusters are available in the following types:

- A local cluster with a single node that you can use to quickly onboard projects.
- A fully-managed cluster that Informatica creates, manages, and deletes.
- A self-service Kubernetes cluster that your organization manages.
- An advanced cluster that a serverless runtime environment creates.

When you start running mappings in advanced mode, you can use a local cluster. As soon as you run a mapping in advanced mode, Data Integration automatically creates a temporary, local cluster to run the mapping so that you can access the mapping output immediately.

As you develop larger data integration projects, your administrator can configure your runtime environment to connect to a fully-managed or self-service cluster that adapts to the size of the workload. Or, your administrator can set up a serverless runtime environment that includes an advanced cluster for your organization to use.

For more information about advanced clusters, see the Administrator help.

Mapping configuration in advanced mode

Use the Mapping Designer to create a mapping in advanced mode and update advanced mode settings.

You can complete the following tasks:

- Create a mapping in advanced mode.
- Copy a mapping to advanced mode.
- Update advanced mode settings.

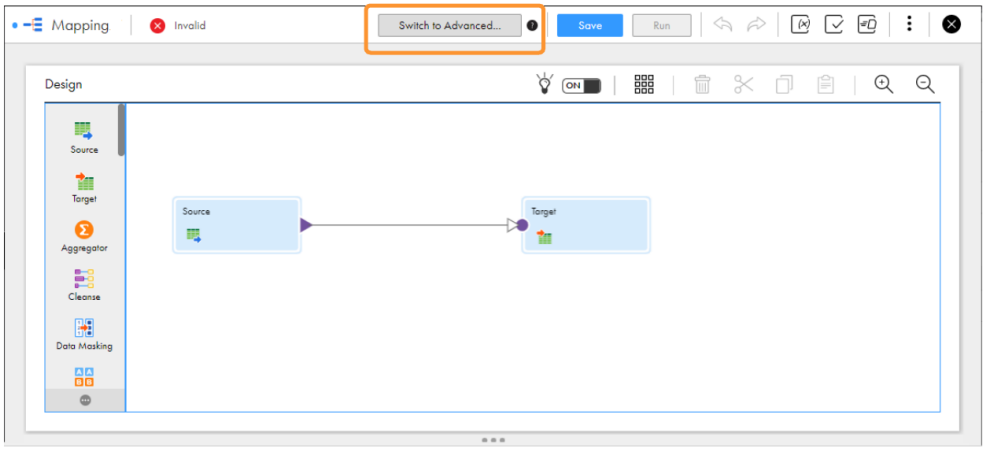
Note: After the Mapping Designer updates the mapping canvas to use advanced mode, you cannot revert the operation.

Creating a mapping in advanced mode

To create a mapping in advanced mode, create a mapping and then update the mapping canvas in the Mapping Designer.

1. In Data Integration, click **New** > **Mappings** > **Mapping**.
2. In the Mapping Designer, click **Switch to Advanced**.

The following image shows the **Switch to Advanced** button in the Mapping Designer:



3. In the **Switch to Advanced** dialog box, optionally choose to hide the advanced mode confirmation dialog box or choose to always create mappings in advanced mode.
4. Click **Switch to Advanced**.

   The Mapping Designer updates the mapping canvas to advanced mode.

Copying a mapping to advanced mode

To copy an existing mapping to advanced mode, update the mapping canvas in the Mapping Designer.

1. Open a mapping.
2. In the Mapping Designer, click **Switch to Advanced**.
3. In the **Switch to Advanced** dialog box, optionally choose to hide the advanced mode confirmation dialog box or choose to always create mappings in advanced mode.
4. Click **Switch to Advanced**.

Data Integration retains the original mapping and creates a copy of the mapping in advanced mode. In the copy, the Mapping Designer updates the mapping canvas to advanced mode.

Updating advanced mode settings

Update the advanced mode settings to show or hide the **Switch to Advanced** dialog box or to create mappings in advanced mode by default.

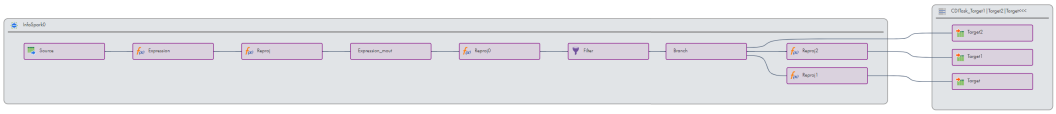
1. Open a mapping in advanced mode.
2. In the Mapping Designer header, click the **Settings** icon.
3. In the **Advanced Mode Settings** dialog box, check or uncheck the following options:
   - Ask for confirmation before copying a mapping to advanced mode.
   - Always create mappings in advanced mode.
4. Click **Save**.

Runtime plans

To troubleshoot a mapping in advanced mode, you can view the runtime plan when you monitor the job. The runtime plan shows which mapping logic runs on the Data Integration Server or on an advanced cluster.

When you run a mapping in advanced mode, Data Integration uses the mapping compilation log to create a visualization of the mapping logic at runtime. The visualization is available as a runtime plan that you can view in the job details.

The following image shows an example of a runtime plan:



The runtime plan is a close approximation of how the data actually flows through the mapping at runtime. The runtime plan is generated after mapping compilation, so the transformations in the runtime plan might appear in a different order than in the designed mapping.

Note: If a mapping runs using SQL ELT optimization, the mapping logic runs on the database, so a runtime plan isn't available.

For information about monitoring a mapping in advanced mode, see *Monitor*.

Mappings in SQL ELT mode

Create a mapping in SQL ELT mode when your source and target are in the same cloud ecosystem and you want all the mapping logic to be processed by your cloud ecosystem. For example, you need to move data in an Azure Data Lake Storage data lake to your Snowflake cloud data warehouse, and you want all the data processing to occur within Snowflake.

When you run a mapping in SQL ELT mode, the transformation logic is translated into ecosystem-specific commands and SQL statements that run in the underlying cloud infrastructure. Because data isn't moved out of the cloud infrastructure to transform it, the overall processing speed increases.

Mappings in SQL ELT mode can read data from a cloud data warehouse and write it to the same cloud data warehouse. They can also read data from a data lake in your cloud ecosystem and write it to a cloud data warehouse in the same ecosystem.

Mappings in SQL ELT mode can extract data from the following sources:

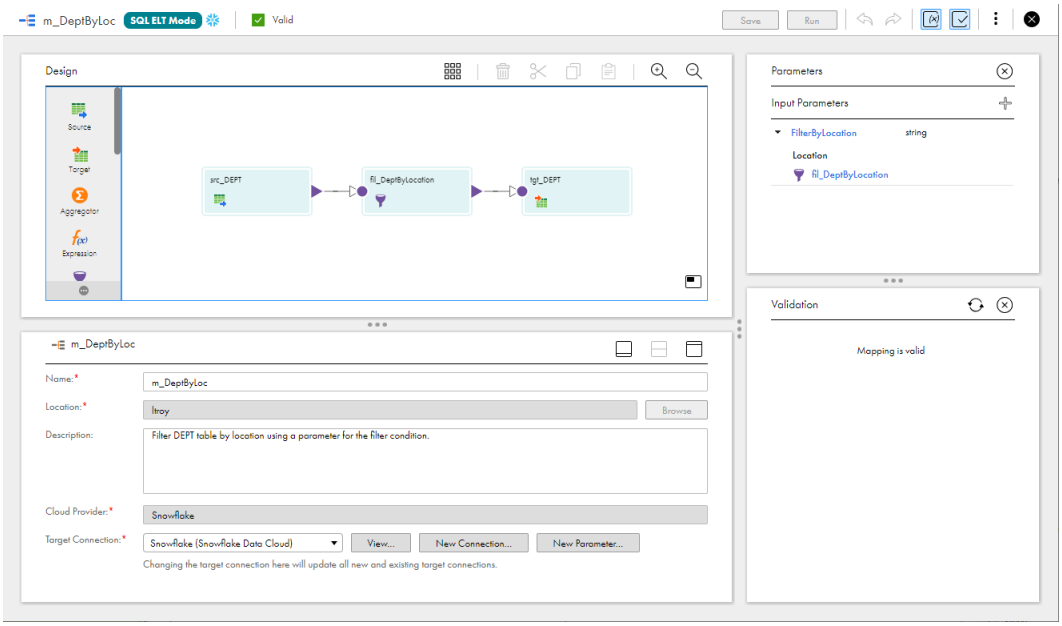
- Snowflake Data Cloud
- Amazon S3
- Microsoft Azure Data Lake Storage Gen2

They can load data to Snowflake Data Cloud.

To create a mapping in SQL ELT mode, you create a mapping and select **Mapping - SQL ELT** as the mapping type. You're then prompted to choose a Snowflake target connection. If your organization doesn't have any Snowflake connections, you're prompted to create one.

After you choose the target connection, the Mapping Designer opens.

The following image shows the Mapping Designer for a mapping in SQL ELT mode:



The mapping is automatically configured with the target connection you chose. You can add additional sources, targets, and transformations to the mapping. The transformations available in the transformation palette are transformations that the target cloud data warehouse can process. For example, Snowflake can't process the logic in a Java transformation, so the transformation palette for mappings in SQL ELT mode doesn't include the Java transformation.

To run a mapping in SQL ELT mode, you create a mapping task, just as you do for any other type of mapping.

Sources in mappings in SQL ELT mode

Sources in mappings in SQL ELT mode can behave differently than sources in other types of mappings.

The sources can behave differently in the following ways:

- When you create a mapping in SQL ELT mode, you can read data from cloud data lakes and cloud data warehouses within the cloud ecosystem. Therefore, when you configure the source connection in the Source transformation, you can choose only an Amazon S3, Azure Data Lake Storage Gen2, or Snowflake Data Cloud connection.
- If you add multiple sources to the mapping, all sources use the same source connection or connection parameter. If you change the source connection or connection parameter in one of the sources, the Mapping Designer updates the other sources so that they all use the same connection.
- Mappings in SQL ELT mode that read data from a cloud data warehouse always write data to the same cloud data warehouse. Therefore, if the source is a Snowflake cloud data warehouse, it must use the same connection or connection parameter as the target.
- When you create a connection parameter in the Source transformation, you can set the connection type to Amazon S3 or Azure Data Lake Storage Gen2. If you want to create a Snowflake connection parameter, you need to create the parameter in the mapping properties where you choose the target connection.
- If you change the Snowflake connection in the mapping properties, the Snowflake connection in the source is automatically updated to use the same connection. If the Source transformation uses an Amazon S3 or Azure Data Lake Storage Gen2 connection, then the source connection doesn't change when you change the target connection.

There might be other behavior differences for sources based on the connector type. For more information, see the *Snowflake Data Cloud Connector* guide.

Targets in mappings in SQL ELT mode

Targets in mappings in SQL ELT mode can behave differently than targets in other types of mappings.

The targets can behave differently in the following ways:

- When you create a mapping in SQL ELT mode, you choose the target connection first. The target connection must be a Snowflake Data Cloud connection.
- You change the target connection in the mapping properties. In the Target transformation, the target connection is read-only.
- If you add multiple targets to the mapping, all targets must use the same target connection or connection parameter. If you change the target connection, the Mapping Designer updates all targets in the mapping to use the new connection.

There might be other behavior differences for targets based on the connector type. For more information, see the *Snowflake Data Cloud Connector* guide.

Transformations in mappings in SQL ELT mode

Transformations in mappings in SQL ELT mode can behave differently than transformations in other types of mappings. The transformations available also differ based on the mapping mode.

A mapping in SQL ELT mode includes only transformations that the cloud ecosystem can process. If you paste an unsupported transformation into the Mapping Designer, the mapping validation fails. For more information about the transformations you can use in a mapping in SQL ELT mode, see *Transformations*.

The following types of transformations behave differently in mappings in SQL ELT mode:

Lookup transformations

When you create a mapping in SQL ELT mode, you can perform a lookup using data from cloud data lakes and cloud data warehouses within the cloud ecosystem. Therefore, when you configure the connection in the Lookup transformation, you can choose only an Amazon S3, Azure Data Lake Storage Gen2, or Snowflake Data Cloud connection.

You can add multiple Lookup transformations to the mapping, but all lookups must use the same connection or connection parameter as the Source transformation. If you change the source connection, the mapping designer updates all Lookup transformations in the mapping so that they use the same connection as the Source transformation.

When you create a connection parameter in a Lookup transformation, you can set the connection type to Amazon S3 or Azure Data Lake Storage Gen2. If you want to create a Snowflake connection parameter, you need to create the parameter in the mapping properties where you choose the target connection.

If you change the Snowflake connection in the mapping properties, the Snowflake connection in the Lookup transformations are automatically updated to use the same connection. If the Lookup transformation uses an Amazon S3 or Azure Data Lake Storage Gen2 connection, then the Lookup source connection doesn't change when you change the target connection.

Sequence transformations

All Sequence transformations need to be connected directly to one downstream Expression transformation or to one target.

Transformations that include expressions

When you configure an expression in a transformation like an Aggregator, Expression, or Filter transformation, the functions you use to create the expression are the cloud ecosystem functions. For example, if you configure an expression in an Expression transformation in a mapping in SQL ELT mode, the functions listed in the expression editor are Snowflake functions, not Informatica transformation language functions.

The expression syntax must be valid in the cloud ecosystem. The expression editor doesn't validate the expression syntax. If you enter an invalid expression, the mapping fails at run time.

There might be other behavior differences for transformations based on the connector type. For more information, see the *Snowflake Data Cloud Connector* guide.

Mapping behavior differences

Mapping behavior can vary based on whether the mapping is a mapping in SQL ELT mode or another type of mapping. The Mapping Designer also differs slightly.

The Mapping Designer for mappings in SQL ELT mode is separate from the Mapping Designer for other mappings. If you want to switch to or from SQL ELT mode, create a new mapping.

Mappings in SQL ELT mode differ from other mappings in the following ways:

- When you create a mapping in SQL ELT mode, you choose the target connection first. To change the target connection, click the mapping canvas and update the mapping properties. You can't change the target connection in the Target transformation.
- If you create expressions within a mapping in SQL ELT mode, you use your cloud data warehouse's native functions and expression syntax instead of Informatica functions and expression syntax. For more information about the native functions and expression syntax, see the documentation for your cloud data warehouse.
- You can't test run a mapping in SQL ELT mode from within the Mapping Designer. To run a mapping in SQL ELT mode, create a mapping task.
- Mappings in SQL ELT mode can't transform hierarchical data. Any field that contains hierarchical data is treated as a pass-through field in all transformations. If you need to transform hierarchical data, create a mapping or mapping in advanced mode.
- Mappings in SQL ELT mode can't include mapplets.
- Mappings in SQL ELT mode can't have multiple flows or pipelines. If you need to process data in multiple pipelines, create a mapping or mapping in advanced mode.
- CLAIRE recommendations and the **Inventory** panel aren't available in mappings in SQL ELT mode.

There might be other mapping behavior differences based on the connector type. For more information, see the *Snowflake Data Cloud Connector* guide.

Mapping configuration in SQL ELT mode

Use the Mapping Designer to create a mapping in SQL ELT mode.

The Mapping Designer for mappings in SQL ELT mode is different than the Mapping Designer for other types of mappings. Therefore, you can't copy mappings in SQL ELT mode to other modes, and you can't copy other types of mappings to SQL ELT mode.

Creating a mapping in SQL ELT mode

To create a mapping in SQL ELT mode, choose **Mapping - SQL ELT** when you create a mapping.

If your organization uses the unified **Home** page and you've configured a primary cloud data warehouse, the target connection is automatically set to your primary cloud data warehouse, but you can change it if you wish. If your organization doesn't use the unified **Home** page or you haven't configured a primary cloud data warehouse, you'll need to choose the cloud platform when you create the mapping.

1. In Data Integration, click **New** > **Mappings** > **Mapping**.
2. Click **Create**.
3. In the **New Mapping** dialog box, select **Mapping - SQL ELT** and then click **Continue**.

   If you've configured a primary cloud data warehouse, the Mapping Designer opens in SQL ELT mode. The target connection is set to your primary cloud data warehouse.
4. If you haven't configured a primary cloud data warehouse, perform the following steps:
   a. Choose your cloud platform and then click **Select**.

      Data Integration prompts you to choose the target connection or create a new one.
   b. Select a target connection.

      The **Target Connection** list displays the target connections for the cloud platform you selected. If your organization has no target connections for the selected platform, or if you want to use a connection that doesn't appear in the drop-down list, click **Add Connection** to create a connection.
   c. Click **Next**.

      The Mapping Designer opens in SQL ELT mode. The target connection is configured with the connection you selected.
5. Configure the target object, configure the source, and add any required transformations to the mapping.
6. Click **Save** to save the mapping.

Automapping

Automapping is a CLAIRE capability that provides recommendations to automatically match and align data integration sources and targets during mapping design.

You can use the automapping capabilities to generate suggestions for data integration mapping in Business 360 Console.

Automapping generates a set of mapping suggestions based on matching data fields' names, structures, and data types. You can also add transformations to the mapping as necessary.

You can use automapping to generate data integration mappings faster when compared to the traditional method of creating mappings. You can then focus on prioritizing the business requirements, while CLAIRE efficiently handles the complex task of generating the data integration mappings.

Using automapping, you can automate the mapping process by learning from the matched records and considering the appropriate mapping rules or transformations, and combine the data from different sources. Automapping reduces the associated complexities of creating mappings by automating the manual process and in turn, reduces the possibilities of errors associated with a manual process.

Automapping, with the help of CLAIRE, can learn the underlying structure and patterns in a mapping to identify source table relationships. Automapping identifies matching fields between source and target transformation, establishes logical mappings between the source and target metadata, and then suggests

the mappings. You can use these suggestions to create a mapping without prior domain knowledge of the different source and target systems.

Automapping isn't available in mappings in SQL ELT mode.

Generate automapping

In Business 360 Console, when you start the mapping creation process from an organization or persona business entity, the fields of source and target objects are mapped automatically in Data Integration.

After you start automapping in Business 360 Console, you are automatically redirected to Data Integration.

1. In Business 360 Console, on the **Sources** tab of the **Choose sources and targets** page, click **Add Sources**.
2. Select the packages and sources along with their objects.

   You can select multiple objects. Use the **Find** field to search for the objects if required.
3. Click **Add**.

   The selected sources and their objects appear on the **Sources** tab.
4. Optionally, verify the target business entity.

   Note that the target is predefined and you can't configure it.
5. Click **Create Mapping**.

   The mapping appears in the mapping canvas. Save and validate the mapping.
   You can configure the mapping to add one or more supported transformations.

Transformations in automapping

You can configure the following transformations in automapping:

- Source
- Target
- Hierarchical target (MDM)
- Filter
- Expression
- Normalizer
- Joiner

For more information about the transformations, see the *Transformation* help.

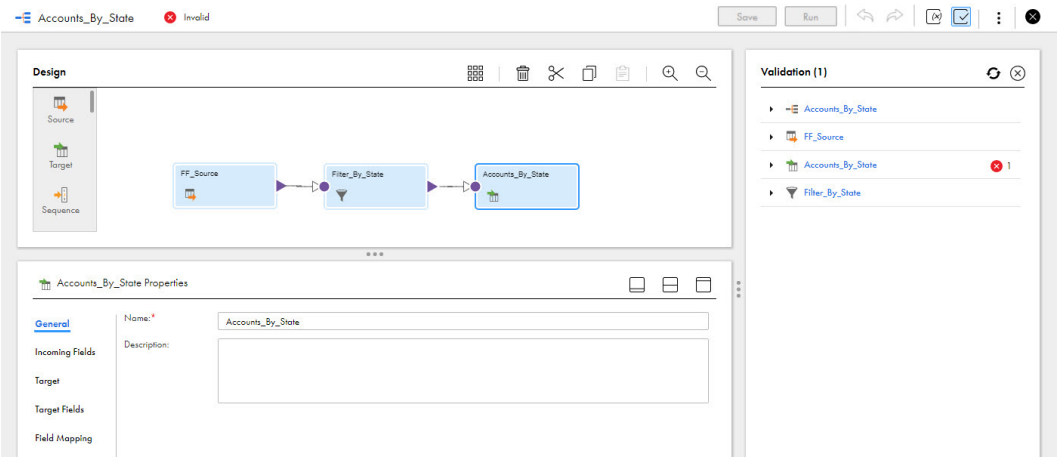
Mapping validation

Each time you save a mapping, the Mapping Designer validates the mapping.

When you save a mapping, check the status to see if the mapping is valid. Mapping status displays in the header of the Mapping Designer.

If the mapping is not valid, you can use the **Validation** panel to view the location and details of mapping errors. The **Validation** panel displays a list of the transformations in the mapping. Error icons display by the transformations that include errors.

In the following example, the Accounts_By_State Target transformation contains one error:



Tip: If you click a transformation name in the **Validation** panel, the transformation is selected in the Mapping Designer.

Validating a mapping

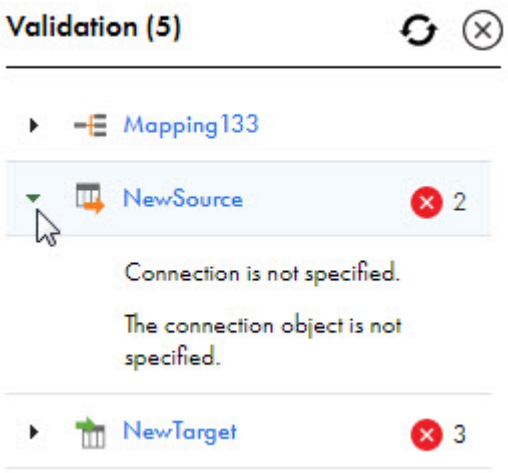
Use the **Validation** panel to view error details.

1. To open the **Validation** panel, in the toolbar, click **Validation** as shown in the following image:



   Error icons appear by the transformations that include errors.

2. To view the list of errors for a transformation, click the down arrow, as shown in the following image:



3. To refresh the **Validation** panel after you make changes to the mapping, click the refresh icon.

Validating a mapping in advanced mode

In advanced mode, Data Integration validates the mapping based on how an advanced cluster processes the data. When you copy a mapping to advanced mode, the **Validation** panel shows validation errors that the advanced cluster checks for.

Validation errors can result from the following differences in advanced mode:

Transformation differences

The transformation palette uses a different set of transformations, so some transformations might no longer be available. Some transformations have a different set of properties that you need to configure, or previously-configured properties might no longer apply.

You might need to reconfigure properties, rearrange the transformations, or replace transformations in the data flow. For example, the Sorter transformation can only take effect in advanced mode when it is directly upstream from the Target transformation.

For more information, see *Transformations*.

Function differences

The expression editor uses a different set of functions. Function return values can differ, and some functions require additional arguments. For example, the TO_DECIMAL function requires the scale argument in advanced mode.

For more information, see *Function Reference*.

Data type differences

Some data types are processed differently. For example, an advanced cluster uses stricter rules to validate and process decimal data types.

For more information, see *Function Reference*.

Connector differences

An advanced cluster accesses some connectors differently than the Data Integration Server. You might need to configure details in the connection properties so that the cluster can access and process the data source.

For more information, see the help for the appropriate connector.

Validating a mapping in SQL ELT mode

In SQL ELT mode, Data Integration validates a mapping using additional rules that don't apply to mappings in other modes.

In SQL ELT mode, validation errors can result due to the following circumstances:

- The mapping contains multiple flows or pipelines.
- The mapping contains a transformation that doesn't appear in the Transformations palette in SQL ELT mode. For example, if you paste an unsupported transformation onto the mapping canvas, a validation error occurs.
- The target connection type or target connection parameter type isn't Snowflake Data Cloud.
- The target has named fields, but the field mapping is set to **Manual**. The field mapping must be **Automatic**, **Partially Parameterized**, or **Completely Parameterized**.
- The source connection or source connection parameter type isn't Snowflake Data Cloud, Amazon S3, or Azure Data Lake Storage Gen2.

- The source and target are in the same cloud data warehouse but they use different connections.
- The mapping contains a Sequence transformation that isn't directly connected to a downstream Expression transformation or target.
- A Sequence transformation is connected to multiple downstream transformations or targets.

Validation errors can also occur when there's a problem with the source, target or lookup connection. For more information, see the *Snowflake Data Cloud Connector* guide.

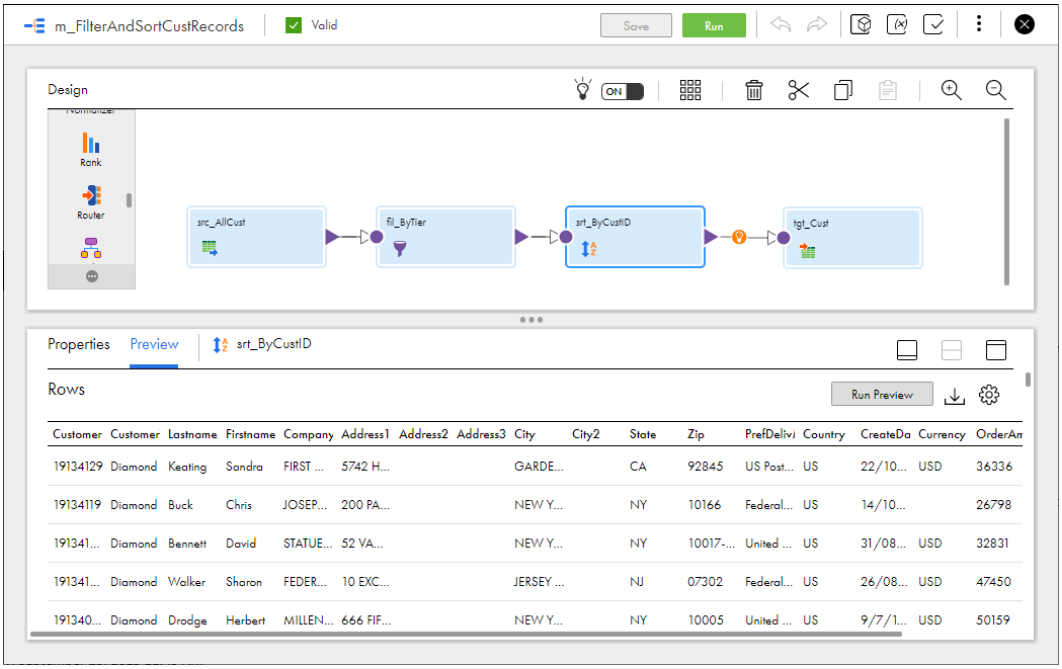
Data preview in mappings

When you create a mapping, you can preview data for individual transformations to test the mapping logic.

You preview data for a transformation on the **Preview** panel of the transformation. Select the number of source rows to process and the runtime environment that runs the preview job.

When you run a preview job, Data Integration creates a temporary mapping task that contains a virtual target immediately downstream of the selected transformation. Data Integration discards the temporary task after the preview job completes. When the job completes, Data Integration displays the data that is transformed by the selected transformation on the **Preview** panel.

The following image shows the **Preview** panel of a Sorter transformation after you run a preview job on the transformation:



To preview data in a mapping, you must have the Data Integration Data Previewer role or your user role must have the "Data - preview" feature privilege for Data Integration.

If you apply a limit to the number of rows to preview for an active transformation, such as a Filter transformation, that row limit also applies to the source. The Filter transformation might show fewer rows output in the filter preview window than in the source preview window, depending on the filter condition.

You can preview data if the mapping uses input parameters. Data Integration prompts you for the parameter values when you run the preview.

You can't preview data when you develop a mapplet in the Mapplet Designer. You can't preview data that contains special, emoji, and Unicode characters in the table name.

Preview behavior for a mapping

You can preview data for a transformation if there are no mapping validation errors in the selected transformation or any upstream transformations. Data Integration displays mapping results for the selected transformation. It also generates results for the upstream transformations.

You can preview data for any transformation except for the following transformations:

- Data Masking
- Hierarchy Builder
- Sequence
- Velocity
- Web Services
- Target

Running a preview job for a mapping

Run a data preview job on the **Preview** panel of the selected transformation.

Before you run a data preview job for a mapping, verify that the following conditions are true:

- Verify that a Secure Agent is available to run the job. You can't run a preview job using the Hosted Agent.
- Verify that the Secure Agent machine has enough disk space to store the preview data.
- Verify that there are no mapping validation errors in the selected transformation or any upstream transformation.

To run a data preview job:

1. In the Mapping Designer, select a transformation.
2. Open the **Preview** panel.
3. Click **Run Preview**.
4. In the Run Preview wizard, enter the number of source rows to preview and the runtime environment to run the preview job.

   The number that you enter applies to each source in the mapping. For example, if you select 10 rows and the mapping contains multiple sources, the preview job processes the first 10 rows in each source.

   You can select up to 999,999,999 rows.

   Warning: Selecting a large number of source rows can cause storage or performance issues on the Secure Agent machine.
5. If the part of the mapping that you are previewing uses input parameters, click **Next** and enter the parameter values.
6. Click **Run Preview**.

Data Integration displays the results in the **Preview** panel for the selected transformation.

You can monitor preview jobs on the **My Jobs** page in Data Integration and on the **All Jobs** and **Running Jobs** pages in Monitor. Data Integration names the preview job <mapping name>-<instance number>, for example,

MyMapping_1. You can download the session log for a data preview job. However, session logs for data preview jobs for mappings in SQL ELT mode aren't available for download.

To restart a preview job, run the job again on the **Preview** panel. You can't restart a data preview job on the **My Jobs** or **All Jobs** pages.

Viewing preview results for a mapping

Data Integration generates preview results for the selected transformation and the upstream transformations in the mapping. Data Integration stores preview results in CSV files on the Secure Agent machine.

Data Integration displays preview results on the **Preview** panel of the selected transformation and each upstream transformation. Data Integration does not display preview results for downstream transformations.

If a transformation has multiple output groups and you want to preview results for a different output group, select the output group from the Output Groups menu at the top of the **Preview** panel.

To download preview results, click **Download** on the **Preview** panel.

Data Integration stores preview results in CSV files on the Secure Agent machine. When you run a preview, Data Integration creates one CSV file for the selected transformation and one CSV file for every upstream transformation in the mapping. If a transformation has multiple output groups, Data Integration creates one CSV file for each output group. If you run the same preview multiple times, Data Integration overwrites the CSV files.

By default, the files are stored in the following directory:

```
<Secure Agent installation directory>/apps/Data_Integration_Server/data/cache/preview
```

The CSV files are stored in this directory unless an organization administrator changes the value of the $PMCacheDir property for the Data Integration Server service that runs on the Secure Agent. For more information about Secure Agent services, see the Administrator help.

Note: Ensure that the Secure Agent machine has enough disk space to store preview data for all users that might run a data preview using the Secure Agent.

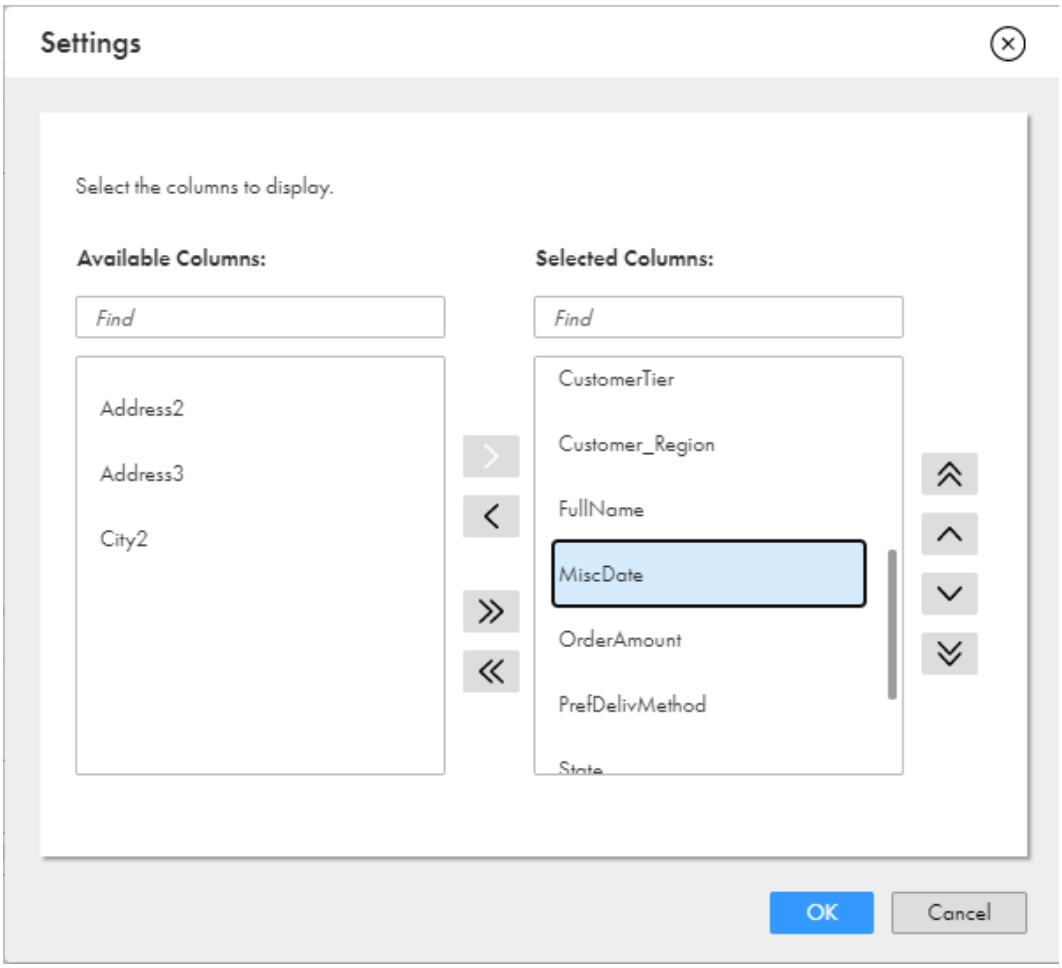
Data Integration purges the preview directory once every 24 hours. During the purge, Data Integration deletes files that are more than 24 hours old.

Note: In CSV format, null values for integer, double, string, and text data types show as empty.

Customizing preview results

You can choose which columns to display on the **Preview** panel. You can also reorder the columns. Customize the **Preview** panel on the **Settings** dialog.

The following image shows the **Settings** dialog:



To open the **Settings** dialog, click the settings icon on the **Preview** panel. Columns in the Selected Columns area appear on the **Preview** panel. To hide a column from the **Preview** panel, select it and move it to the Available Columns area. To reorder the columns in the **Preview** panel, select a column name in the Selected Columns area and move it up or down.

Data preview in advanced mode

When you create a mapping in advanced mode, you can preview data for individual transformations to test the mapping logic. The preview job runs on an advanced cluster in an AWS or Azure cloud environment and uses on-demand instances.

You can preview data if there are no mapping validation errors in the selected transformation. A mapping in advanced mode shows preview data for the selected transformation.

You can't preview data for the following transformations:

- Sources that have binary fields
- Normalizer
- Router
- Hierarchy Processor with multiple output groups
- Target

Note: If the cluster is running many jobs, the preview job might be slow.

Running a preview job in advanced mode

Run a data preview job on the **Preview** panel of the selected transformation.

Before you run a data preview job, verify that there are no mapping validation errors in the selected transformation.

1. In the Mapping Designer, select a transformation.
2. Open the **Preview** panel.
3. Click **Run Preview**.
4. In the Run Preview wizard, enter the number of source rows to preview and the runtime environment to run the preview job.

   The number that you enter applies to each source in the mapping. For example, if you select 10 rows and the mapping contains multiple sources, the preview job processes the first 10 rows in each source.

   You can select up to 999,999,999 rows.

   Warning: Selecting a large number of source rows can cause storage or performance issues on the Secure Agent machine.
5. Select the **Read Entire Source** option if you're sure you want Data Integration to read all source data. Preview data may be slow to display if you have a large source data set. Target data will be limited based on the value entered in the Rows to Preview field.
6. The **Enable Upstream Preview** option allows you to preview data for the selected transformation and upstream transformations. This option is enabled by default. Preview data may be slow to display if you have many upstream transformations.**
7. Click **Run Preview**.

   Data Integration displays the results in the **Preview** panel for the selected transformation.

You can monitor preview jobs on the **My Jobs** page in Data Integration and on the **All Jobs** and **Running Jobs** pages in Monitor. Data Integration names the preview job <mapping name>-<instance number>, for example, MyMapping_1. You can download the session log for a data preview job.

To restart a preview job, run the job again on the **Preview** panel. You cannot restart a data preview job on the **My Jobs** or **All Jobs** pages.

Note: If you preview data that has parameterized sources or transformations with complex fields on a local cluster, you can only specify an S3/ADLS staging directory.

Viewing preview results in advanced mode

Data Integration generates preview results for the selected transformation and stores preview results in the staging location for the advanced cluster.

Data Integration shows complex data cells as hyperlinks on the **Preview** page. When you click a hyperlink, the preview data appears in a separate panel. Preview results show null values for the job name, start time, end time, and job status columns.

Preview results for the Sequence transformation might not be consistent from one preview run to the next. For example when you set the initial value in the transformation to 1, you might get a result of 1, 2, 3, 4 the first time you run a preview. The second time you run a preview, you might get 2001, 2002, 2003, 2004.

If the selected transformation is available only in advanced mode or contains hierarchical data, Data Integration stores preview results in a JSON file. Otherwise, Data Integration stores preview results in a CSV file. Data Integration purges preview data every 30 minutes and deletes preview data when the cluster shuts down.

Note: When the preview results are in CSV format, null values for integer, double, string, and text data types show as empty.

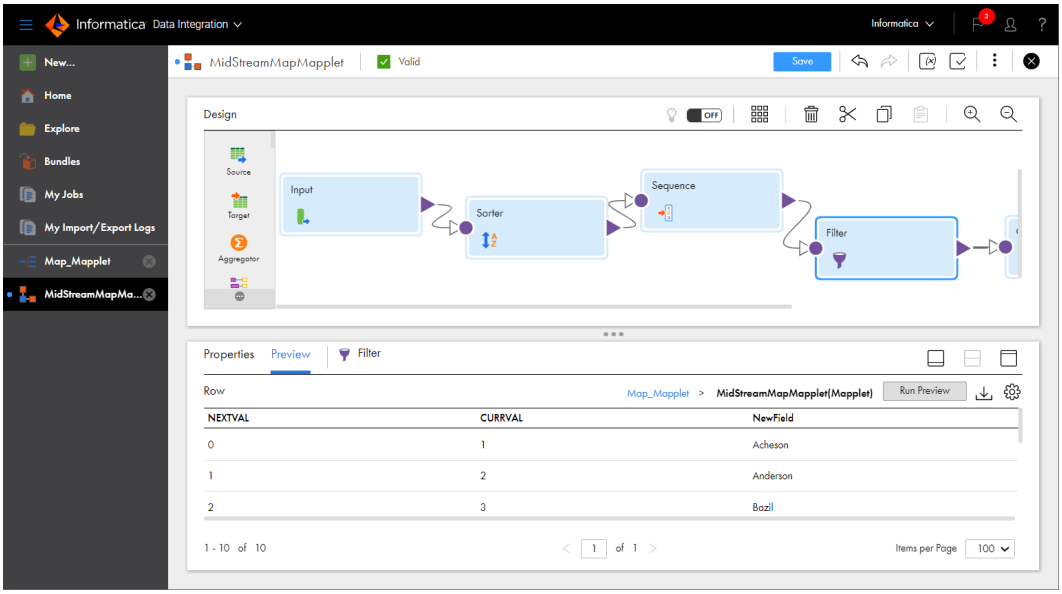
Detailed data preview in mapplets

When you preview data in a mapping that contains a Mapplet transformation, you can drill down on the Mapplet transformation and preview data for transformations in the mapplet. If the mapplet contains another Mapplet transformation, you can continue to drill down to subsequent mapplets.

When you preview transformations in a mapplet, Data Integration runs data preview in the context of the mapping that you drilled down from. Data Integration creates a temporary preview job that uses the mapping source data and a virtual target created directly downstream from the transformation that you want to preview in the mapplet. After the job runs, Data Integration displays data that is transformed by the selected transformation.

To preview data in a mapplet, you must drill down to the mapplet from a mapping even if the mapplet contains a Source transformation. You cannot run data preview in a mapplet that you open in the Mapplet Designer.

The following image shows the **Preview** panel after running data preview in a mapplet:



You run a preview job in a mapplet the same way that you run a preview job in a mapping. The **Data Preview** wizard displays the preview context of the mapplet as a breadcrumb and the mapping that the mapplet is referenced in. If the mapping or mapplet contains input parameters, Data Integration prompts you to enter parameter values when you run the preview.

You can preview data for any transformation except for the following transformations:

- Data Masking
- Hierarchy Builder
- Input
- Output
- Sequence
- Velocity
- Web Services
- Target

You can only preview mapplet data if the mapplet was created in Data Integration and is referenced in a mapping that was created after the April 2022 release.

Running detailed data preview in a mapplet

When you run a data preview job in a mapping that contains a Mapplet transformation, you can drill down on the Mapplet transformation and preview data as it passes through the mapplet.

Before you run a detailed preview for a mapplet, verify that the following requirements are met:

- The mapping is valid.
- The mapping was created after the April 2022 release.
- The Mapplet transformation uses a mapplet created in Data Integration.

To preview detailed data in a mapplet, perform the following steps:

1. In the mapping, open the preview panel for the Mapplet transformation.

2. Click **Detailed Preview in Mapplet**.

   The mapplet opens.
3. In the mapplet, select the transformation that you want to preview and click **Run Preview**.
4. In the **Data Preview** wizard, configure the number of rows to preview, the runtime environment, and any parameters.
5. Click **Run Preview**.
6. To return to the mapping or to another mapplet, click the asset name in the breadcrumb.

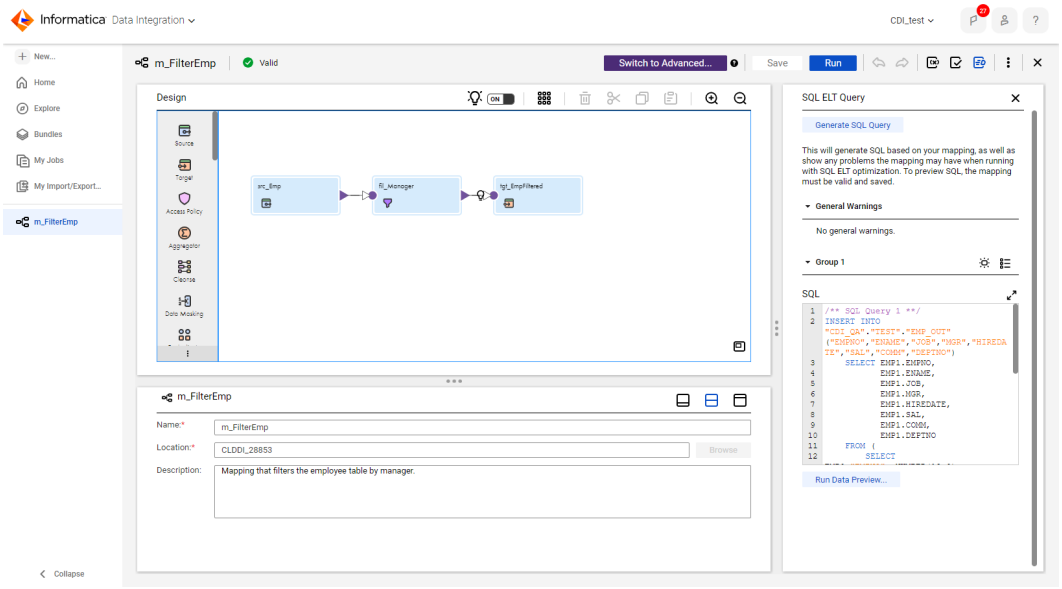
SQL ELT query preview

When you create a mapping that is configured for SQL ELT optimization or a mapping in SQL ELT mode, you can preview the SQL queries that Data Integration pushes to the database or cloud data warehouse. Preview these queries in the **SQL ELT Query** panel in the Mapping Designer.

You can preview SQL ELT queries for some connector types. For more information, see the help for the appropriate connector.

When you preview SQL ELT queries, Data Integration creates and runs a temporary SQL ELT preview mapping task. When the job completes, Data Integration displays the SQL to be executed and any warnings in the **SQL ELT Query** panel.

The following image shows the **SQL ELT Query** panel:



For mappings that are configured for SQL ELT optimization, Data Integration groups SQL and warnings based on the data flow run order. After you run the SQL ELT preview job, you can preview the data after it's transformed by the transformations included in the SQL queries.

If the SQL ELT optimization type that you select isn't available, Data Integration lists the SQL queries, if any, that can be executed. For example, if you select Full SQL ELT optimization, but the target doesn't support it, Data Integration displays the SQL queries that will be pushed to the source.

You can't preview SQL ELT queries in advanced mode.

Running an SQL ELT query preview for mappings configured for SQL optimization

Preview the SQL queries that Data Integration pushes to the database on the **SQL ELT Query** panel.

Before you run an SQL ELT query preview, verify that the following conditions are true:

- In-out parameters have a default value. You can't provide values for in-out parameters when you configure the preview job.
- The mapping is valid.

To run an SQL ELT query preview for a mapping that is configured for SQL optimization:

1. Open the **SQL ELT Query** panel.
2. Click **Generate SQL Query**.
3. In the **SQL ELT Preview** wizard, select the runtime environment and then click **Next**.
4. If the mapping contains input parameters, enter the parameter values and then click **Next**.
5. Configure SQL ELT optimization options.
6. Click **SQL ELT Preview**.

Data Integration displays the SQL queries and any warnings in the **SQL ELT Query** panel. If a warning relates to a transformation, click the warning to select the transformation on the mapping canvas. If SQL ELT optimization fails, Data Integration lists any queries generated up to the point of failure.

To view the transformations included in SQL ELT optimization, click **View Transformations** for the group you want to view. To highlight the group on the mapping canvas, click **Highlight** for the group you want to see.

You can monitor preview jobs on the **My Jobs**, **Running Jobs**, and **All Jobs** pages. Data Integration names the job <mapping name>_sql_elt_preview-<instance number>, for example, Mapping1_sql_elt_preview-2. You can download the session log for the preview job.

If you update the mapping after you run the preview job, the preview is no longer valid. To restart a preview job, run it again from the **SQL ELT Query** panel. You can't restart the job from the **My Jobs**, **Running Jobs**, or **All Jobs** page.

Running an SQL ELT query preview for mappings in SQL ELT mode

Preview the SQL queries that Data Integration pushes to the cloud data warehouse on the **SQL ELT Query** panel.

Before you run an SQL ELT query preview, verify that the mapping is valid.

To run an SQL ELT query preview for a mapping in SQL ELT mode:

1. Open the **SQL ELT Query** panel.
2. Click **Generate SQL Query**.
3. If the mapping contains input parameters, enter the parameter values and then click **SQL ELT Preview**.

Data Integration displays the SQL queries and any warnings in the **SQL ELT Query** panel. If a warning relates to a transformation, click the warning to select the transformation on the mapping canvas. If query generation fails at any point, Data Integration lists any queries generated up to the point of failure.

You can monitor preview jobs on the **My Jobs**, **Running Jobs**, and **All Jobs** pages. Data Integration names the job <mapping name>_sql_elt_preview-<instance number>, for example, Mapping1_sql_elt_preview-2.

If you update the mapping after you run the preview job, the preview is no longer valid. To restart a preview job, run it again from the **SQL ELT Query** panel. You can't restart the job from the **My Jobs**, **Running Jobs**, or **All Jobs** page.

SQL ELT query preview results files

Data Integration stores preview results in a JSON file on the Secure Agent machine.

If you run a preview more than once, Data Integration overwrites the JSON file.

By default, files are stored in the following directory:

```
<Secure Agent installation directory>/apps/Data_Integration_Server/data/cache/pdo_preview
```

Files are stored in this directory unless the organization administrator changes the $PMCacheDir property for the Data Integration Server service on the Secure Agent. For more information about Secure Agent services, see the Administrator help.

Data Integration purges the directory once every 24 hours. During the purge, Data Integration deletes files that are more than 24 hours old.

SQL ELT query data preview

After you run an SQL ELT query preview for a mapping that is configured for SQL ELT optimization, you can preview the data after it is transformed by the transformation logic included in the queries. You can preview the first 15 rows of data for any group in the **SQL ELT Query** panel.

To preview the data, you must have the Data Integration Data Previewer role or your user role must have the "Data - preview" feature privilege for Data Integration.

When you run an SQL ELT query preview, Data Integration creates a temporary mapping task that contains a virtual target immediately before the target or the last transformation possible for SQL ELT optimization.

When you run SQL ELT query data preview, Data Integration runs a data preview job using the parameter values and session attributes that you entered when you configured the SQL ELT query preview job. The transformation that Data Integration runs data preview on depends on the type of SQL ELT optimization that is possible.

The following table describes the transformation that data preview runs on for each type of optimization:

Optimization type	Transformation
Full	Transformation immediately before the Target transformation.
Partial	Last transformation where SQL ELT optimization is possible.

You can't preview SQL ELT query data for the following transformations:

- Data Masking
- Hierarchy Builder
- Router
- Sequence
- Velocity
- Web Services
- Target

You can't preview SQL ELT query data for a mapping if the last transformation where SQL ELT optimization is possible is a Mapplet transformation. You can't preview the data for mappings in SQL ELT mode.

If you run an SQL ELT query data preview job at the same time you run a mapping data preview job in the same mapping, the job fails.

Running an SQL ELT query data preview job

To preview the data after the SQL queries are executed, run data preview. You run SQL ELT query data preview after you run SQL ELT query preview.

You can't run SQL ELT query data preview jobs for mappings in SQL ELT mode.

1. In the group that you want to preview data for, click **Run Data Preview**.
2. When the job completes, click **View Data Preview**.

   The **SQL ELT optimization data preview** window opens.
3. To download the data preview results as a CSV file, click **Download Preview Results**.

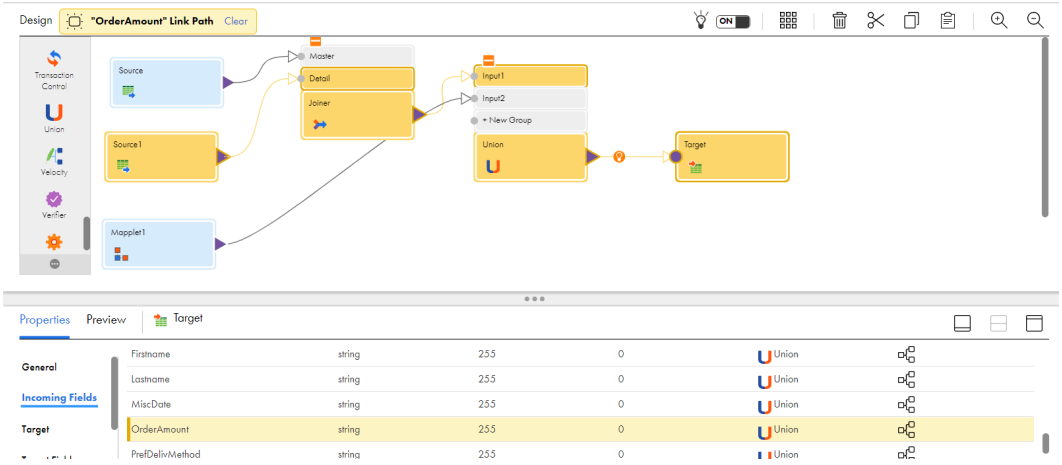
Field lineage

You can view the lineage of an individual field in a mapping. A field's lineage shows how the field is created, renamed, mapped, or changed within each transformation in the data pipeline.

You might want to view a field's lineage to help you troubleshoot a mapping with incorrect target data. For example, if fields are missing in the target, you can trace each field's lineage from its source to find where the field is excluded.

When you view the lineage for a field, Data Integration highlights the field's path on the mapping canvas. Depending on the transformation, it also highlights the field on the **Incoming Fields** and **Output Fields** tabs of the selected transformation. To see how the field moves through the highlighted data flow, select another highlighted transformation.

The following image shows the lineage of the incoming field OrderAmount in the Target transformation:



Data Integration displays the full field lineage. A field's lineage begins at its source and ends at its target. If you add a field midstream, the transformation where you add it is the source. For example, lineage for an expression field begins at the Expression transformation where the field was added. A field's lineage ends at the transformation that does not output the field. For example, if an incoming field is not mapped to a normalized field in a Normalizer transformation, the field lineage stops at the Normalizer transformation.

A field's lineage can depend on the transformation you view the lineage from. For example, when you view the lineage for an incoming field used in the lookup condition of a Lookup transformation, the downstream lineage includes all fields returned by the lookup. If you view the lineage of the same field from a downstream transformation, the upstream lineage does not include the fields returned by the lookup.

You can view the lineage of fields that pass through or are changed by any transformation in the pipeline except the following transformations:

- Hierarchy Builder
- Hierarchy Parser
- Hierarchy Processor
- Java
- Machine Learning
- Mapplet transformations that reference an SAP or PowerCenter mapplet
- Python
- SQL transformations that process an SQL query
- Unconnected transformations

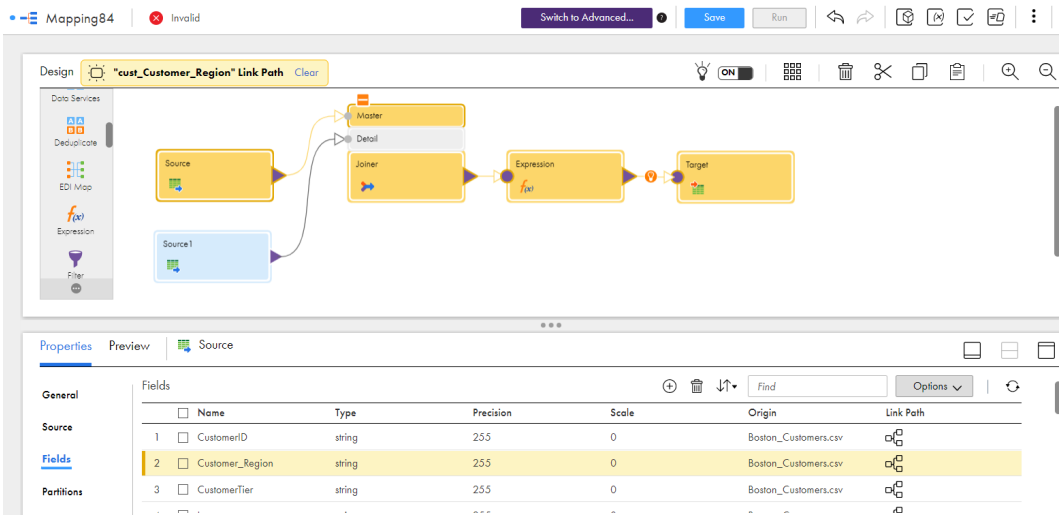
If the mapping contains hierarchical data, you can view the field lineage for primitive fields in the data flow. You can also view the lineage for primitive fields that are output from a hierarchical expression. The upstream lineage includes the hierarchical field but does not include its nested elements.

Lineage for renamed fields

When you view the lineage for a field that was renamed, the upstream lineage includes instances of the field with the original name and the new name.

For example, you have a mapping that joins customer and order tables that have some common field names. In the Joiner transformation, you resolve the field name conflict by prefixing fields in the customer table with cust_ and then configure the transformation to join data based on the customer ID.

In the Target transformation, you want to view the lineage for the cust_Customer_Region field. Data Integration includes the Customer_Region source field in the lineage as shown in the following image:



Lineage for fields containing expressions

You can view the lineage for expression fields, aggregate fields, and for advanced filters that contain incoming fields.

When the expression contains incoming fields, the upstream lineage includes the fields used in the expression.

For example, you configure the expression field Full_Name to concatenate the Firstname and Lastname source fields before loading the data to the target. When you view the field lineage of the expression field Full_Name, the upstream lineage includes the Firstname and Lastname fields.

If the expression doesn't contain incoming fields, the lineage includes only downstream transformations. Lineage also doesn't include variable fields used in the expression.

You can view the upstream field lineage for an advanced filter condition if the expression contains one or more incoming fields. Data Integration doesn't include the downstream lineage because Filter transformations do not include output fields. If you want to view the full lineage for a field included in a filter condition, view the lineage from an upstream or downstream transformation.

Lineage for mapped fields

When you view the lineage for a field that is mapped to another field, the lineage includes both the incoming and output fields.

For example, you have a Normalizer transformation that normalizes incoming quarterly sales data. You create the normalized field Y with an occurs value of 4.

In the Normalizer transformation, you configure the following field mapping:

Incoming field	Normalized field
Q1	Y_1
Q2	Y_2
Q3	Y_3
Q4	Y_4

The normalized field Y is then passed to the downstream transformation.

On the **Incoming Fields** tab of the downstream transformation, you get the lineage for field Y. Because fields Q1, Q2, Q3, and Q4 are mapped to occurrences of Y, they are included in the upstream lineage of Y.

Mapplet field lineage

You can view the lineage of fields that pass through a Mapplet transformation when the transformation references a mapplet that you created in Data Integration.

If the field has at least one lineage path through the mapplet, Data Integration highlights the lineage on the **Incoming Fields** and **Output Fields** tabs of the Mapplet transformation. You cannot drill down to the mapplet to view field lineage. If you want to see how the field is transformed within the mapplet, open the mapplet and view the field lineage in the Mapplet Designer.

If the mapplet was changed, synchronize the mapplet before viewing field lineage in the Mapping Designer. Synchronizing the mapplet ensures that you get the most up-to-date lineage.

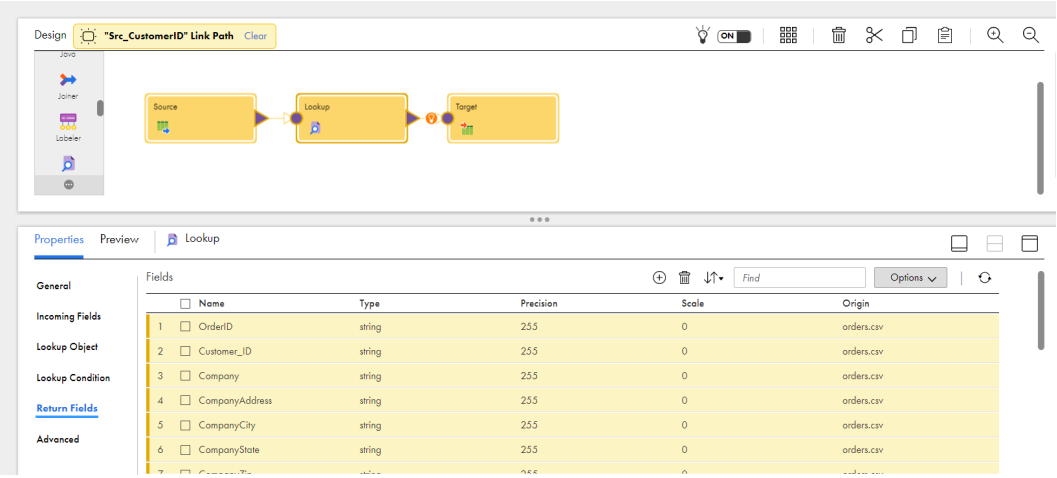
If the Mapplet transformation references an SAP or PowerCenter mapplet, field lineage stops at the Mapplet transformation.

Lineage for lookup fields

When you view the lineage of a field used in the lookup condition of a Lookup transformation, the downstream lineage includes all fields returned by the lookup.

For example, you have a source table with customer data and you want to augment the data with data from an orders table before loading the data to a new target table. You configure the Lookup transformation to return fields when the source field Src_CustomerID equals the lookup field Customer_ID. On the **Incoming Fields** tab of the Lookup transformation, you view the lineage for the Src_CustomerID field. The downstream lineage includes all fields returned by the lookup.

The following image shows the lineage for the Src_CustomerID field on the **Returned Fields** tab:



Lineage for transformations that read Data Quality assets

A transformation in a mapping may read an asset that originates in Data Quality.

Field lineage for transformations that read Data Quality assets behaves in the same manner as other transformations in Data Integration, with the following exceptions:

Rule Specification transformations

When you view the field lineage for a mapped or unmapped output field from a Rule Specification transformation on the target transformation, the field lineage stops at the Rule Specification transformation.

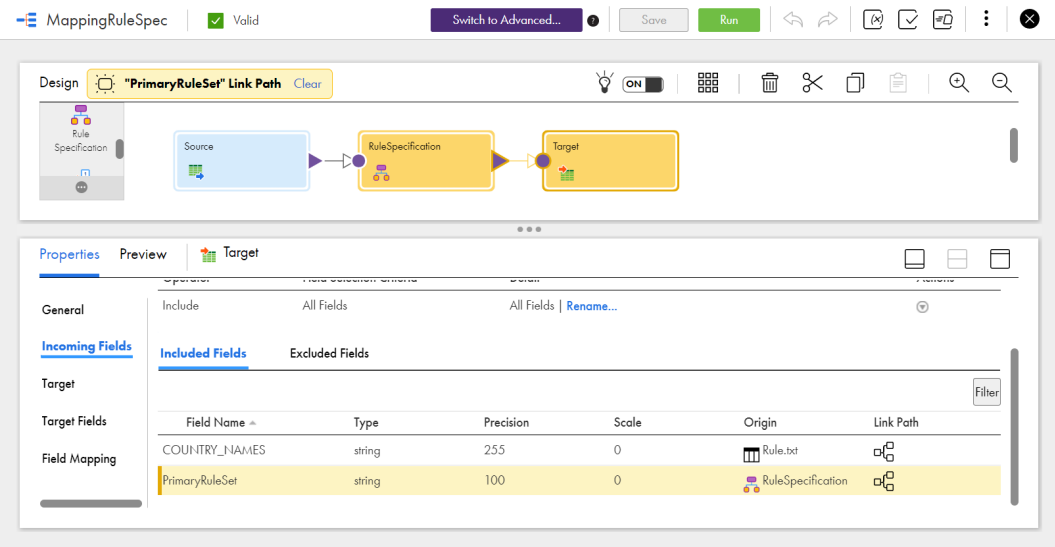
Cleanse transformations

When you view the field lineage for a merged output field from a Cleanse transformation on the target transformation, the field lineage stops at the Cleanse transformation.

Deduplicate transformations

When you view the field lineage for a metadata field from a Deduplicate transformation on the target transformation, the field lineage stops at the Deduplicate transformation.

The following image shows the lineage of the output field in the target transformation:



Viewing field lineage

Depending on the transformation, you can view field lineage from the **Incoming Fields** or the transformation-specific properties tabs.

Before you view a field's lineage, resolve any parameters in the mapping that include the field you are investigating. Data Integration does not include parameters in field lineage.

1. In the row that contains the field that you want to investigate, click **Link Path**.

   Data Integration highlights the field's path on the mapping canvas and highlights the field on the properties tab.

2. Click an upstream or downstream transformation to see how it transforms the field.

   Data Integration automatically opens the relevant properties tab and highlights the field lineage. Depending on the transformation, Data Integration highlights instances of the field on the **Output Fields** and **Incoming Fields** tabs.

3. Select another transformation in the field's path and view instances of the field in the **Properties** panel.

4. When you are finished, click **Clear** in the upper-left corner of the mapping canvas.

Testing a mapping

After you complete a mapping and you confirm that the mapping is valid, you can perform a test run to verify the results of the mapping. Perform a test run of a valid mapping to verify the results of the mapping before you create a mapping task.

When you perform a test run, you run a temporary mapping task. The task reads source data, writes target data, and performs all calculations in the data flow. Data Integration discards the temporary task after the test run.

You can perform a test run from the Mapping Designer or from the **Explore** page.

To test run a mapping from the Mapping Designer, perform the following steps:

1. After you save the mapping, click **Run**.
2. Select the runtime environment and specify values for parameters, if any parameters are in mapping.
3. Click **Run**.

To test run a mapping from the **Explore** page, perform the following steps:

1. Navigate to the mapping and in the row that contains the mapping, click **Actions** and select **Run**.
2. Select the runtime environment and then click **Run**.

Note that if you select **New Mapping Task** instead of **Run**, Data Integration creates a mapping task and saves it in the location you specify. For more information about mapping tasks, see *Tasks*.

You can't test run a mapping in SQL ELT mode. To run a mapping in SQL ELT mode, create a mapping task.

Mapping maintenance

You can view, configure, copy, copy to advanced mode, move, delete, and test run mappings from the **Explore** page.

When you use the **View** action to look at a mapping, the mapping opens in the Mapping Designer. You can navigate through the mapping and select transformations to view the transformation details. You cannot edit the mapping in View mode.

When you copy a mapping, the new mapping uses the original mapping name with a number appended. For example, when you copy a mapping named ComplexMapping, the new mapping name is ComplexMapping_2. When you copy a mapping to advanced mode, the new mapping uses the original mapping name with `Advanced` appended.

You can delete a mapping that is not used by a mapping task. Before you delete a mapping that is used in a task, delete the task or update the task to use a different mapping.

Mapping revisions and mapping tasks

You might need to update a mapping that is used in a mapping task.

When you update a mapping that is used in a mapping task and the mapping is valid, the changes are deployed to the mapping task. If the mapping is invalid, the changes are not deployed to the mapping task and the task uses the valid version of the mapping.

If you change the mapping so that the mapping task is incompatible with the mapping, an error occurs when you run the mapping task. For example, you add a parameter to a mapping after the mapping task was created and you do not update the mapping task to specify a value for the parameter. When you run the mapping task, an error occurs.

If you do not want your updates to affect the mapping task, you can make a copy of the mapping, give the new mapping a different name, and then apply your updates to the new mapping.

Bigint data conversion

In mappings created before the Spring 2020 September release, Data Integration converts Bigint data from a parameterized source to Int data in database targets created at runtime. To write Bigint data to the target without conversion, edit the mapping in the Mapping Designer. Click **Actions** > **Advanced Properties** and enable the option in the mapping advanced properties.

Data Integration does not convert Bigint data in mappings created after the Spring 2020 September release.

CHAPTER 2

Mapping tutorial

The following tutorial describes how to create a simple mapping, save and validate the mapping, and create a mapping task.

For this tutorial, you have a file that contains U.S. customer account information. You want to create a file that contains customer account information for a specific state.

In the mapping, you define the following components:

- Source transformation. Represents the source file that contains account information.
- Filter transformation. Transformation that filters out all account information except for a specific state.
- Target transformation. Represents the target file that contains account information for a specific state.

You create parameters in the mapping so that you can use the same mapping task to create files for each state. You create the following parameters:

- In the Filter transformation, you create a parameter to hold the state value so that you can use the mapping to create multiple tasks. You can specify a different state value for each task.
- In the Target transformation, you create a parameter for the target object so that you have separate target files for each state.

After you define the mapping, you create a mapping task that is based on the mapping. When you run the mapping task, you specify values for the target object parameter and the filter parameter. The mapping task then writes the data to the specified target based on the specified state.

The mapping tutorial includes the following steps:

1. Create a mapping.
2. Configure a source. Specify the name of the source object and the connection to use.
3. Create a Filter transformation. Create a parameter in the Filter transformation to hold the state value.
4. Configure a target. Specify the connection to use and create a parameter for the target object.
5. Validate the mapping to be sure there are no errors. The mapping must be valid before you can run a task based on the mapping.
6. Create a task based on the mapping. The task includes parameters for the target object and state value, which you specify when you run the task.

Preparing for the mapping tutorial

Before you start the mapping tutorial, you need to download a sample file and set up a connection.

1. Download the sample Account source file from the Informatica Cloud Community and save the file in a directory local to the Secure Agent. You can download the file from the following link:

   Sample Source File for the Mapping Tutorial

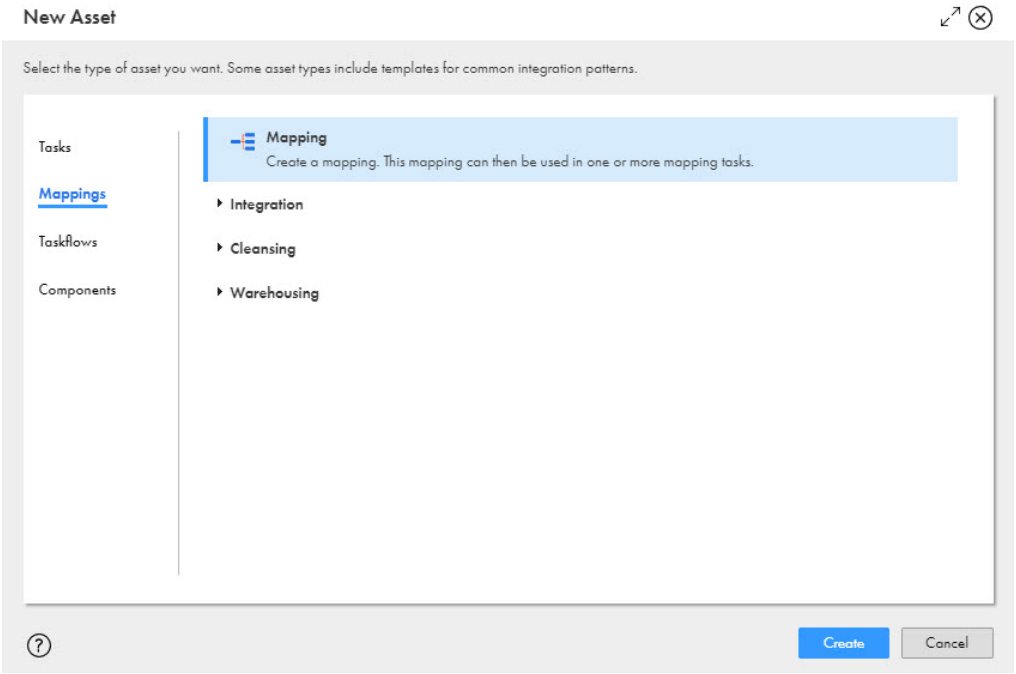
2. Create a flat file connection for the directory that contains the saved sample Account source file.
3. On the **Explore** page, create a project and name the project `AccountsByState`, and then create a folder in the project and name the folder `Mappings`.

Step 1. Create a mapping

In the following procedure, you create a mapping in the Mapping Designer to specify the source, filter, and target.

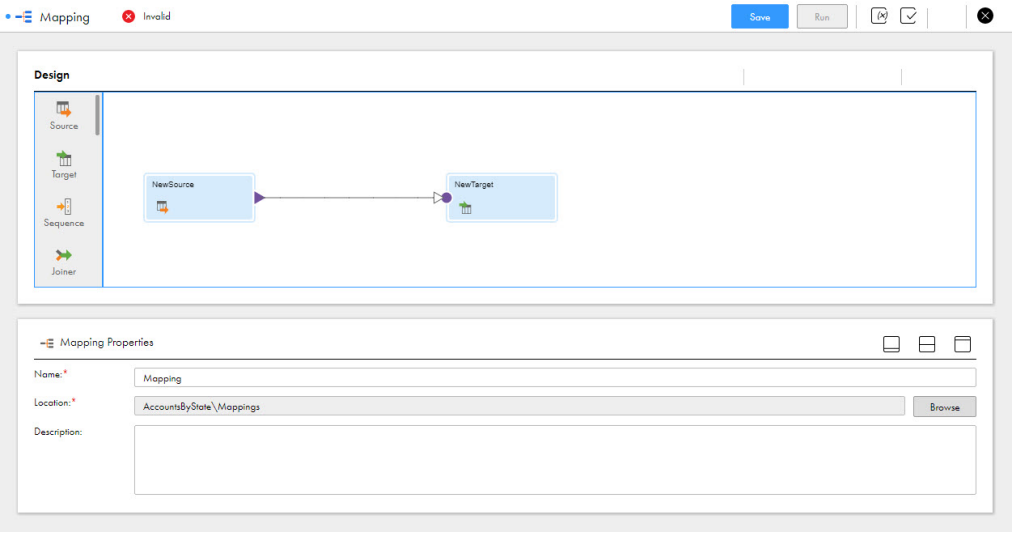
1. To create a mapping, click **New** and then in the **New Asset** dialog box, click **Mapping**.

   The following image shows the **New Asset** dialog box with **Mapping** selected:



2. Click **Create**. The Mapping Designer appears with a new mapping displayed in the mapping canvas.

The following image shows a new mapping in the Mapping Designer:



3. In the **Properties** panel, enter `m_Accounts_by_State` for the mapping name.

   You can use underscores in mapping and transformation names, but do not use other special characters.

   Tip: Informatica recommends that you use a standard naming convention for objects. For example, begin all object names with an abbreviation of the object type so that mapping names begin with m_, Source transformation names begin with src_, parameter names begin with p_, and so on. Also, use names that explain the purpose of the object, for example, flt_Filter_by_State. A standard naming convention is particularly helpful when you are working with large, complex mappings so that you can easily identify the type and purpose of each object.



4. To select a location for the mapping, browse to the folder you want the mapping to reside in, or use the default location.

   If the **Explore** page is currently active and a project or folder is selected, the default location for the asset is the selected project or folder. Otherwise, the default location is the location of the most recently saved asset.

Step 2. Configure a source

In the following procedure, you create a Source transformation to specify the source object.

When you design a mapping, the first transformation you configure is the Source transformation. You specify the source object in the Source transformation. The source object represents the source of the data that you want to use in the mapping. You add the source object at the beginning of mapping design because the source properties can affect the downstream data. For example, you might filter data at the source, which affects the data that enters downstream transformations.

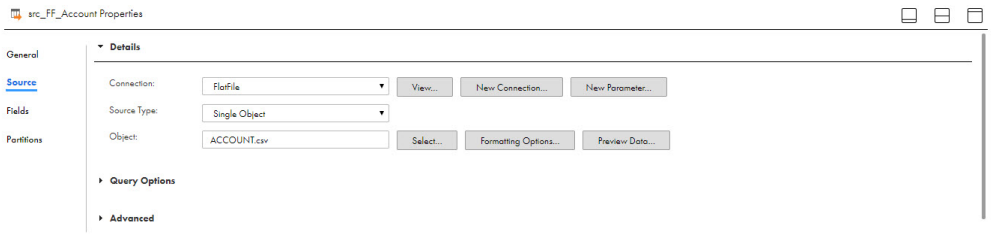
Configure the sample Account flat file as the source object.

1. On the mapping canvas, click the Source transformation to select it.
2. In the Properties panel, click **General** and enter src_FF_Account for the Source transformation name.
3. Specify which connection to use based on the source object. In this case, the source object is a flat file so the connection needs to be a flat file connection.

   Click **Source** and configure the following properties:

Source detail	Description
Connection	Connection to the sample source file. Select the flat file connection that you set up before you started the tutorial.
Source Type	Source object or a parameter, which is a placeholder for the source object that you specify when you run a task based on the mapping. Select **Single Object**.
Object	Source object for the mapping. Click Select and navigate to the ACCOUNT.csv source file. Or, enter the full path to the source object and the source file name, for example, C:\Informatica\Tutorial\ACCOUNT.csv. If you want to view the data in the source file, you can click **Preview Data**.

   The following image shows the src_FF_Account details in the Properties panel:



   To view the source fields and field metadata, click the **Fields** tab.

4. To save the mapping and continue, click **Save**.

Step 3. Create a Filter transformation

In the following procedure, you create a Filter transformation to filter accounts based on the state in which each account is located. You also create a parameter to hold state values.

You want this mapping to run tasks that filter accounts based on specific states. To accomplish this, you add a Filter transformation to the data flow to capture state information. You then define a parameter in the filter condition to hold the state value. When you use a parameter, you can reuse the same mapping to create multiple tasks. You can specify a different state value for each task. Or you can use the same mapping task and change the value for state when you run the task.

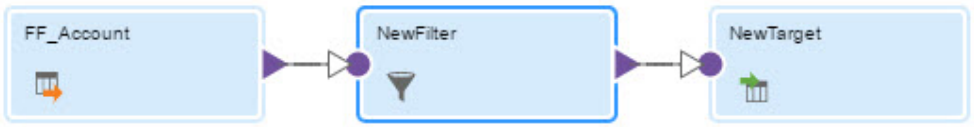
The sample Account source file includes a State field. When you use the State field in the filter condition, you can write data to the target based on the state. For example, when you use State = MD as the condition, you include accounts based in Maryland in the data flow. When you use a parameter for the value of the filter condition, you can define the state that you want to use when you run the task.

Field rules define the fields that enter the transformation and how they are named. By default, all available fields are included in the transformation. You might want to exclude unnecessary fields when you have large source files. Or you might want to change the names of certain incoming fields, for example, when you have multiple sources in a mapping. Field rules are configured on the **Incoming Fields** tab. For this tutorial, do not configure field rules.

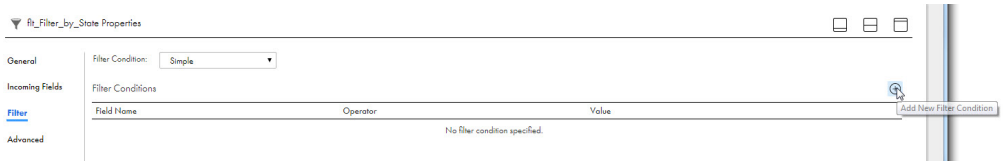
1. To add a Filter transformation, drag a Filter transformation from the Transformation palette to the mapping canvas and drop it between the src_FF_Account Source transformation and the NewTarget transformation.

   Note: You might need to scroll through the Transformation palette to find the Filter transformation.

   When you drop a new transformation in between two transformations in the canvas, the new transformation is automatically linked in the data flow, as shown in the following image:



   When you link transformations, the downstream transformation inherits fields from the previous transformation.
2. To configure the Filter transformation, select the Filter transformation on the mapping canvas.
3. To name the Filter transformation, in the Properties panel, click **General** and enter `flt_Filter_by_State` for the Filter transformation name.
4. To create a simple filter with a parameter for the value, click **Filter**. For the **Filter Condition**, select **Simple**.
5. Click **Add New Filter Condition**, as shown in the following image:

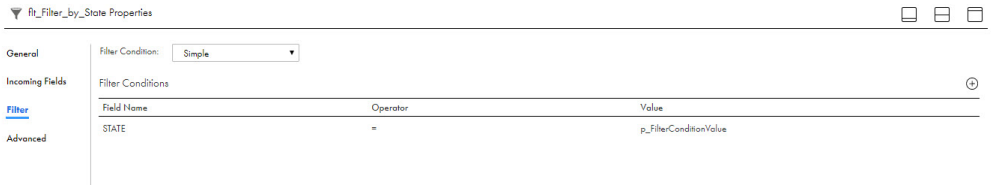


   When you click **Add New Filter Condition**, a new row is created where you specify values for the new filter condition.
6. For **Field Name**, select **State**.
7. For **Operator**, select **Equals**.
8. To parameterize the filter condition value, for **Value**, select **New Parameter**.
9. In the **New Parameter** dialog box, configure the following options:

Filter Condition Detail	Description
Name	Name of the filter condition. Enter `p_FilterConditionValue` for the name.
Display Label	Label that shows in the mapping task wizard where you enter the condition value. Enter `Filter Value for State` for the label.

Filter Condition Detail	Description
Description	Description that appears in the mapping task wizard. Enter this text for the description: `Enter the two-character state name for the data you want to use.`
Type	Datatype of the field used for the filter condition. The State field is a String datatype, which is already specified in the dialog box.
Default Value	Default value for the filter condition. The mapping task uses this value unless you specify a different value. You want to run the task for accounts in Maryland by default, so enter `MD`.

10. Click **OK**. The new filter condition displays in the Properties panel, as shown in the following image:



11. To save your changes, click **Save**.

Step 4. Configure a target

In the following procedure, you create a parameter for the target object so that each time you run the mapping task, you can select a different target.

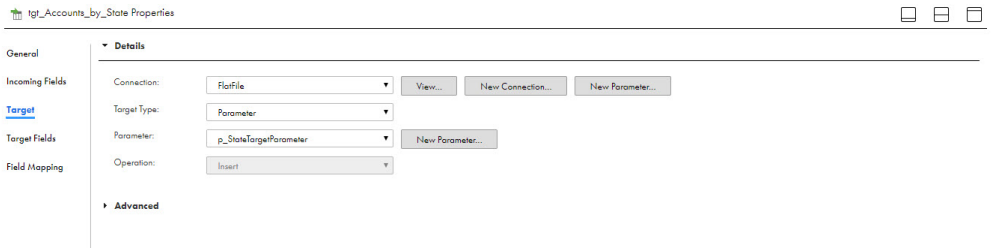
For example, you might want files that only include data for the states you specify in the filter parameter when you run the mapping task. When you run the task, if you filter for accounts in California, you can select the file that contains data for California accounts as the target.

1. On the mapping canvas, click the Target transformation to select it.
2. To name the Target transformation, in the **Properties** panel, click **General** and enter `tgt_Accounts_by_State` for the Target transformation name.

3. Click **Target** and configure the following properties:

Target detail	Description
Connection	Connection to the target file. You want the target object to be a flat file and you want the target object to reside in the same location as the source file. You can use the same connection that you used for the source because the source is a flat file as well.
Target Type	Target object or parameter. You want to parameterize the target object so that you can have separate files for each state, so select **Parameter**.
Parameter	Parameter to use for the target object. This field only appears when you select **Parameter** as the target type. Click **New Parameter**, and for the parameter name, enter `p_StateTargetParameter`. For the display label, enter `Accounts for State`. Click **OK**.

The following image shows the properties for the tgt_Accounts_by_State Target transformation:



4. Click **Field Mapping** and for **Field map options**, select Automatic. You cannot specify field mappings because the target object is parameterized. Because you can select different target objects each time you run the task, the fields in the target objects might not be the same each time you run the task.
5. Click **Save**. You now have a complete mapping.

Step 5. Validate and test the mapping

In the following procedure, you save and validate the mapping and then you test run the mapping.

You can save a mapping that is not valid. However, you cannot run a task that uses a mapping that is not valid. An example of an invalid mapping is a Source transformation or Target transformation that does not have a specified connection, or a mapping that does not have a Source transformation and a Target transformation.

1. To validate the mapping, click **Save**.

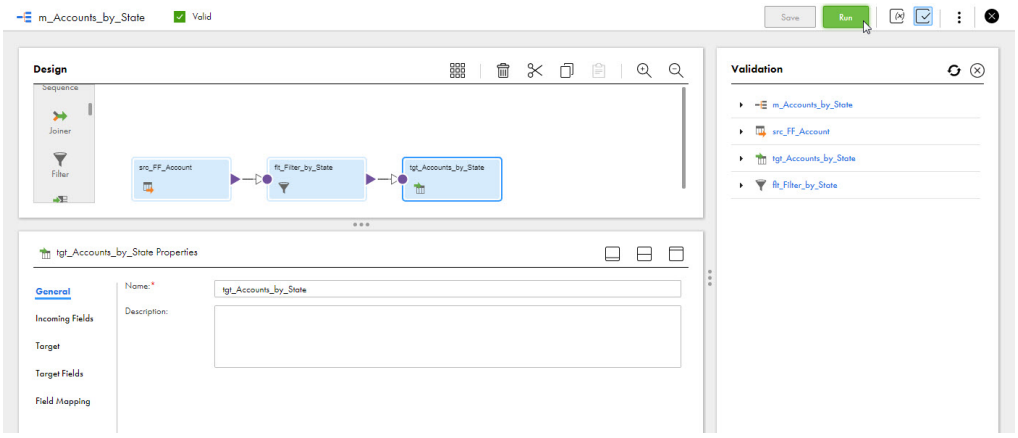
   Whenever you save the mapping, the Mapping Designer validates the mapping. The status of the mapping displays in the header next to the mapping name. The mapping status can be Valid or Invalid.

2. If the mapping is not valid, perform the following steps:
   a. In the header, click the **Validation** icon to open the **Validation** panel.

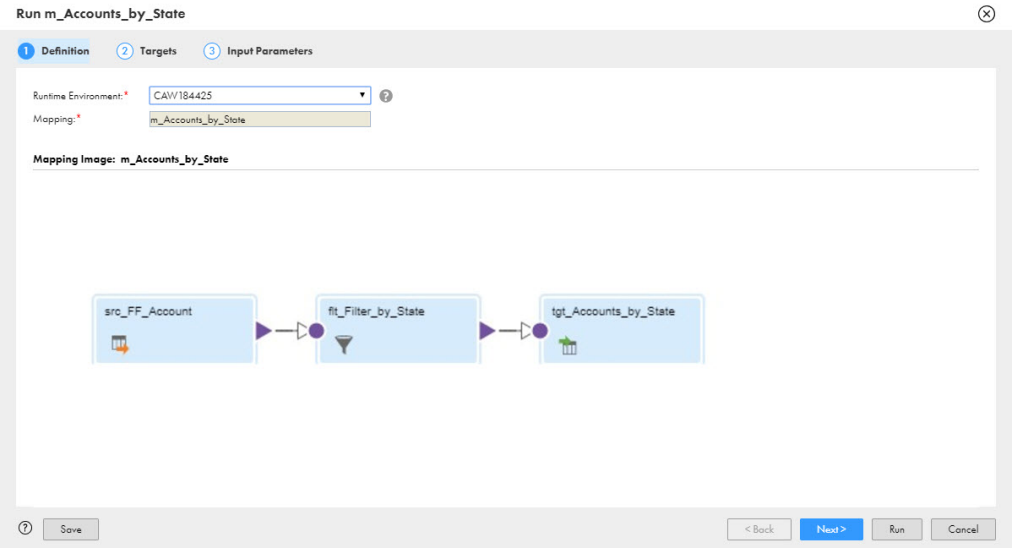
      The **Validation** panel lists the mapping and the transformations used in the mapping and shows where the errors occur. For example, in the following image, the tgt_Accounts_by_State Target transformation has an error:



   b. After you correct the errors, save the mapping and then in the **Validation** panel, click **Refresh**. The **Validation** panel updates to list any errors that might still be present.
3. To test the mapping, click **Run**.



4. In the wizard, select the runtime environment and click **Next**.

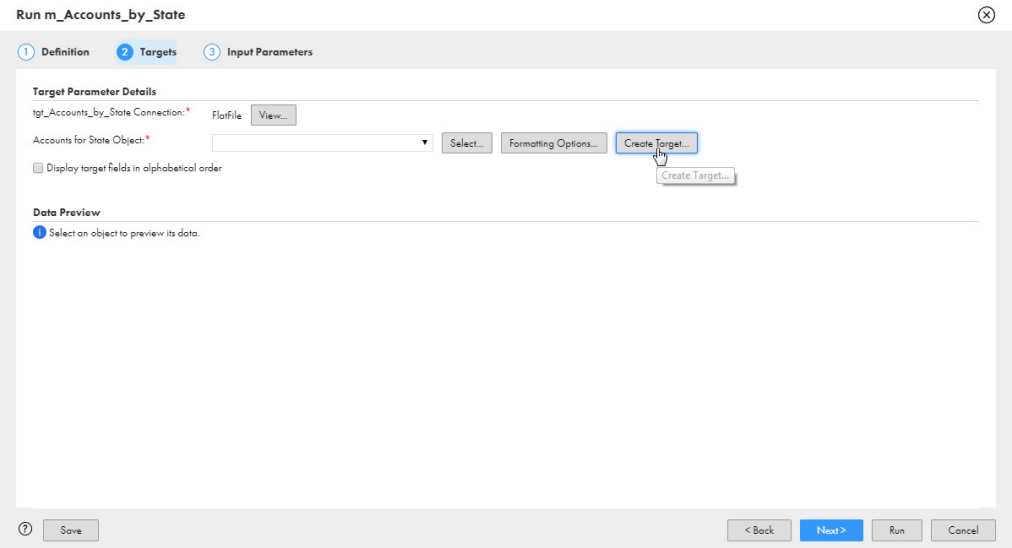


The **Targets** page appears. The **Targets** page appears because the target is parameterized. If you create a mapping that does not include a parameterized target, you will not see the **Targets** page when you run the mapping.

The parameter you created for the target object displays on the page with the Accounts for State label, as you specified when you created the parameter.

You can select a target object or create a new target object. For this tutorial, let's create a new target object that will include accounts for Texas.

5. On the **Targets** page, click **Create Target**.



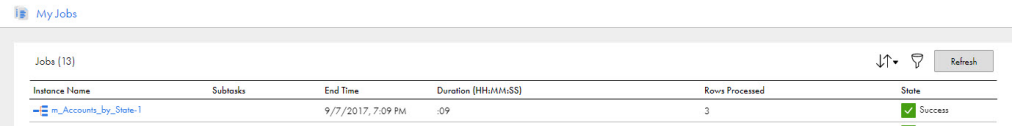
6. Enter the object name `Accounts_By_State_TX` and click **OK**.
7. Click **Next**.

   The **Input Parameters** page appears. In the **Filter Value for State** text box, note that the default value you entered for the filter condition, `MD`, displays in the text box.
8. Change the value from `MD` to `TX` so that the target file will include the accounts from Texas.

9. Click **Run**.

   The mapping task runs and then returns you to the Mapping Designer.

10. In the navigation bar, click **My Jobs**. The **My Jobs** page lists all of the jobs that you have run. At the top, you should see the mapping task that was created when you ran the mapping, as shown in the following diagram:



   The **My Jobs** page shows that there were three accounts from Texas, which are now in your tgt_Accounts_By_State_TX target file.

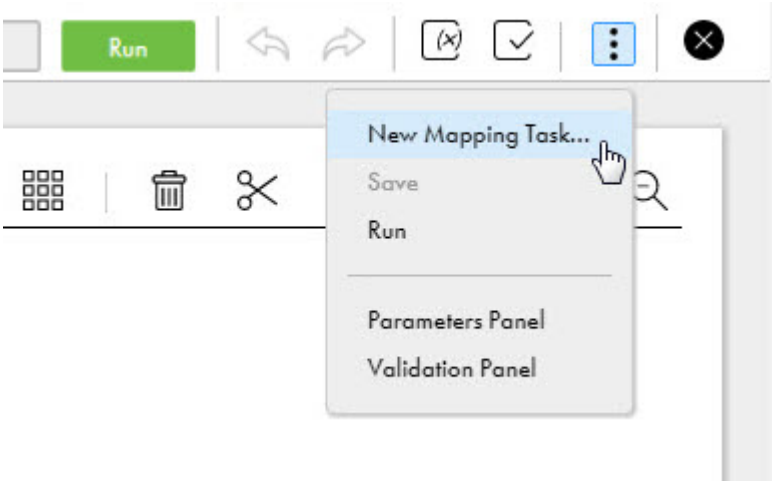
ID2	NAME2	STREET2	CITY2	STATE	ZIPCODE2	CARD_TYP	ANNUAL_REVENUE2
12345	United Oil	17th Aven	South City	TX	52330	Silver	3400000
23422	Edge Com	312 Consti	Austin	TX	34933	Bronze	1.39E+08
59993	Pyramid C	2 Place Ju	Paris	TX	75251	Gold	9.5E+08

Step 6. Create a mapping task

In the following procedure, you create a mapping task that uses the mapping that you just designed.

Now that you have a valid mapping, you can use the mapping task wizard to create tasks based on the mapping. Because you used parameters in the mapping, each time you run the task, you can change the parameter values. After the task runs, you have a target file that contains account information for the state that you specify in the filter condition parameter.

1. To create a mapping task based on the mapping, while still in the Mapping Designer, click **Actions** and select **New Mapping Task**, as shown in the following image:

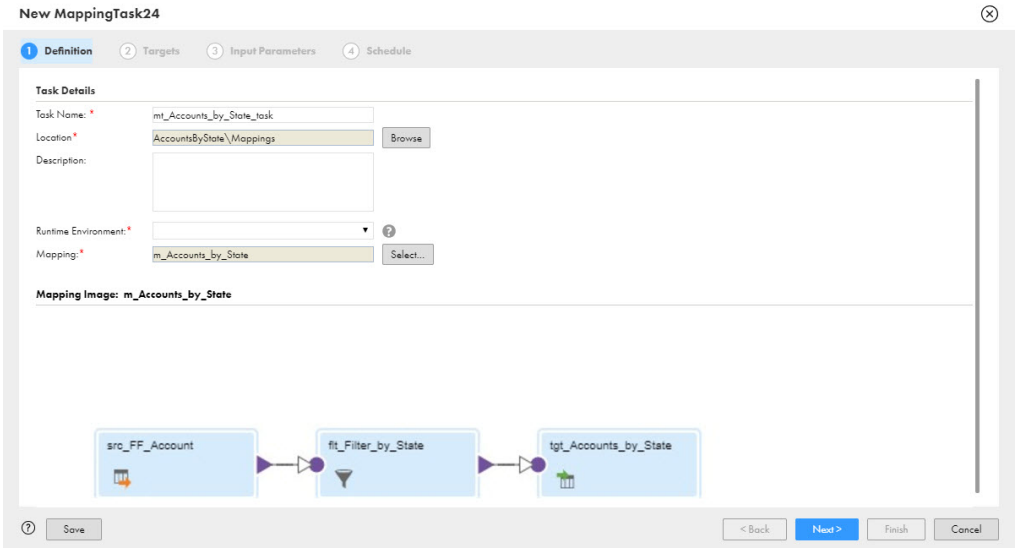


   The mapping task wizard appears.

2. On the **Definition** page, enter `mt_Accounts_by_State_task` for the name of the task.

3. Select the runtime environment to use to run this task. It should be the same runtime environment that you used to create and test the mapping.

   The following image shows the **Definition** page in the mapping task wizard:



4. Click **Next**.

   The **Targets** page appears. You can either select a target object from the list or click **Create Target** to create a new target.

5. Click **Next**.

   The **Input Parameters** page appears and displays the **Filter Value for State** filter condition. You can change the value of the parameter, if desired.

6. Click **Next**.

   The **Schedule** page appears. Because this is a tutorial and we do not want to run this mapping task on a regular basis, leave the default values.

7. Click **Finish** to save the Accounts by State mapping task.

CHAPTER 3

Parameters

Parameters are placeholders that represent values in a mapping or mapplet. Use parameters to hold values that you want to define at run-time such as a source connection, a target object, or the join condition for a Joiner transformation. You can also use parameters to hold values that change between task runs such as a time stamp that is incremented each time a mapping runs.

You can create the following kinds of parameters in a mapping or mapplet:

Input Parameters

An input parameter is a placeholder for a value or values in a mapping or mapplet. Input parameters help you control the logical aspects of a data flow or to set other variables that you can use to manage different targets.

When you define an input parameter in a mapping, you set the value of the parameter when you configure a mapping task.

In-Out Parameters

An in-out parameter holds a variable value that can change each time a task runs, to handle things like incremental data loading. When you define an in-out parameter, you can set a default value in the mapping but you typically set the value at run time using an Expression transformation. You can also change the value in the mapping task.

Input parameters

An input parameter is a placeholder for a value or values in a mapping. You define the value of the parameter when you configure the mapping task.

You can create an input parameter for logical aspects of a data flow. For example, you might use a parameter in a filter condition and a parameter for the target object. Then, you can create multiple tasks based on the mapping and write different sets of data to different targets. You could also use an input parameter for the target connection to write target data to different Salesforce accounts.

The following table describes the input parameters that you can create in each transformation:

Transformation	Input parameter use in mappings and tasks
Source	You can use an input parameter for the following parts of the Source transformation: - Source connection. You can configure the connection type for the parameter or allow any connection type. In the task, you select the connection to use. - Source object. In the task, you select the source object to use. For relational and Salesforce connections, you can specify a custom query for a source object. - Filter. In the task, you configure the filter expression to use. To use a filter for a parameterized source, you must use a parameter for the filter. - Sort. In the task, you select the fields and type of sorting to use. To sort data for a parameterized source, you must use a parameter for the sort options.

Target	You can use an input parameter for the following parts of the Target transformation:
	- Target connection. You can configure the connection type for the parameter or allow any connection type. In the task, you select the connection to use.
	- Target object. In the task, you select the target object to use.
	- Completely parameterized field mapping. In the task, you configure the entire field mapping for the task.
	- Partially parameterized field mapping. Based on how you configure the parameter, you can use the partial field mapping parameter as follows:
	  - Configure links in the mapping and display unmapped fields in the task.
	  - Configure links in the mapping and display all fields in the task. Allows you to edit links configured in the mapping.

All transformations with incoming fields	You can use an input parameter for the following parts of the **Incoming Fields** tab of any transformation: - Field rule: Named field. You can use a parameter when you use the Named Fields field selection criteria for a field rule. In the task, you select the field to use in the field rule. - Renaming fields: Pattern. You can use a parameter to rename fields in bulk with the pattern option. In the task, you enter the regular expression to use.
Aggregator	You can use an input parameter for the following parts of the Aggregator transformation: - Group by: Field name. In the task, you select the incoming field to use. - Aggregate expression: Additional aggregate fields. In the task, you specify the fields to use. - Aggregate expression: Expression for aggregate field. In the task, you specify the expression to use for each aggregate field.
Data Masking	You can use an input parameter for masking techniques in the Data Masking transformation. In the task, you select and configure the masking techniques.
Expression	You can use an input parameter for an expression in the Expression transformation. In the task, you create the entire expression.
Filter	You can use an input parameter for the following parts of the Filter transformation: - Completely parameterized filter condition. In the task, you enter the incoming field and value, or you enter an advanced data filter. - Simple or advanced filter condition: Field name. In the task, you select the incoming field to use. - Simple or advanced filter condition: Value. In the task, you select the value to use.
Joiner	You can use an input parameter for the following parts of the Joiner transformation: - Join condition. In the task, you define the entire join condition. - Join condition: Master field. In the task, you select the field in the master source to use. - Join condition: Detail field. In the task, you select the field in the detail source to use.

Transformation	Input parameter use in mappings and tasks
Lookup	You can use an input parameter for the following parts of the Lookup transformation: - Lookup connection. You can configure the connection type for the parameter or allow any connection type. In the task, you select the connection to use. - Lookup object. In the task, you select the lookup object to use. - Lookup condition: Lookup field. In the task, you select the field in the lookup object to use. - Lookup condition: Incoming field. In the task, you select the field in the data flow to use.

Mapplet	You can use an input parameter for the following parts of the Mapplet transformation:
	- Connection. If the mapplet uses connections, you can configure the connection type for the parameter or allow any connection type. In the task, you select the connection to use.
	- Completely parameterized field mapping. In the task, you configure the entire field mapping for the task.
	- Partially parameterized field mapping. Based on how you configure the parameter, you can use the partial field mapping parameter as follows:
	  - Configure links in the mapping that you want to enforce, and display unmapped fields in the task.
	  - Configure links in the mapping, and allow all fields and links to appear in the task for configuration.
	You can configure input parameters separately for each input group.

Rank	You can use an input parameter for the number of rows to include in each rank group. In the task, you enter the number of rows.
Router	You can use an input parameter for the following parts of the Router transformation: - Completely parameterized group filter condition. In the task, you enter the expression for the group filter condition. - Simple or advanced group filter condition: Field name. In the task, you select the incoming field to use. - Simple or advanced group filter condition: Value. In the task, you select the value to use.
Sorter	You can use an input parameter for the following parts of the Sorter transformation: - Sort condition: Sort field. In the task, you select the field to sort. - Sort condition: Sort Order. In the task, you select either ascending or descending sort order.
SQL	You can use an input parameter for the following parts of the SQL transformation: - Connection: In the Mapping Designer, select the stored procedure or function before you parameterize the connection. Use the **Oracle or SQL Server** connection type. In the task, you select the connection to use. - User-entered query: You can use string parameters to define the query. In the task, you enter the query.

Structure Parser	You can use an input parameter for the following parts of the Structure Parser transformation:
	- Completely parameterized field mapping. In the task, you configure the entire field mapping for the task.
	- Partially parameterized field mapping. Based on how you configure the parameter, you can use the partial field mapping parameter as follows:
	  - Configure links in the mapping that you want to enforce, and display unmapped fields in the task.
	  - Configure links in the mapping, and allow all fields and links to appear in the task for configuration.

Transformation	Input parameter use in mappings and tasks
Transaction Control	You can use an input parameter for the following parts of the Transaction Control transformation: - Transaction Control condition: In the task, you specify the expression to use as the transaction control condition. - Advanced transaction control condition: Expression. In the task, you specify the string or field to use in the expression.

Union	You can use an input parameter for the following parts of the Union transformation:
	- Completely parameterized field mapping. In the task, you configure the entire field mapping for the task.
	- Partially parameterized field mapping. Based on how you configure the parameter, you can use the partial field mapping parameter as follows:
	  - Configure links in the mapping that you want to enforce, and display unmapped fields in the task.
	  - Configure links in the mapping, and allow all fields and links to appear in the task for configuration.
	You can configure input parameters separately for each input group.

Input parameter types

You can create different types of input parameters. The type of parameter indicates how and where you can use the parameter.

For example, when you create a connection parameter, you can use it as a source, target, or lookup connection. An expression parameter can represent an entire expression in the Expression transformation or the join condition in the Joiner transformation. In a transformation, only input parameters of the appropriate type display for selection.

You can create the following types of input parameters:

string

Represents a string value to be used as entered.

In the task, the string parameter displays as a text box in most instances. A Named Fields string parameter displays a list of fields from which you can select a field.

You can use string parameters in the following locations:

- All transformations: Field rule bulk rename by pattern
- All transformations: Field name for the Named Fields field selection criteria
- Filter condition value in the Filter transformation
- Joiner condition value in the Joiner transformation
- User-entered query in the SQL transformation
- Advanced transaction control condition in the Transaction Control transformation

connection

Represents a connection. You can specify the connection type for the parameter or allow any connection type.

In the task, the connection parameter displays a list of connections.

You can use connection parameters in the following locations:

- Source connection

- Lookup connection
- Mapplet connection
- Database connection in the SQL transformation
- Target connection

If you want to use a connection parameter with a data object or query, configure the mapping with an actual connection. After you configure the mapping logic, replace the connection with the connection parameter. If you need to edit the object or query, in the mapping, reselect the connection. After you save your changes, replace the connection with the connection parameter again.

expression

Represents an expression.

In the task, displays the Field Expression dialog box to configure an expression.

You can use expression parameters in the following locations:

- Full expression in the Expression transformation
- Full join condition in the Joiner transformation
- Full lookup condition in the Lookup transformation
- Transaction control condition in the Transaction Control transformation

data object

Represents a data object, such as a source table or source file.

In the task, appears as a list of available objects from the selected connection.

You can use data object parameters in the following locations:

- Source object
- Lookup object
- Target object

field

Represents a field.

In the task, displays as a list of available fields from the selected object.

You can use field parameters in the following locations:

- Field in a filter condition in the Filter transformation
- Field in a join condition in the Joiner transformation
- Field in a lookup condition in the Lookup transformation
- Advanced transaction control condition in the Transaction Control transformation

field mapping

Represents field mappings for the task. You can create a full or partial field mapping.

Use a full field mapping parameter to configure all field mappings in the task. In the task, a full field mapping parameter displays all fields for configuration.

Use a partial field mapping to configure field mappings in the mapping and in the task.

You can use a partial field mapping parameter as follows:

- Preserve links configured in the mapping. Link fields in the mapping that must be used in the task. In the task, the parameter displays the unmapped fields.

- Allow changes to the links configured in the mapping. Link fields in the mapping that can be changed in the task.
  In the task, the parameter displays all fields and the links configured in the mapping. You can create links and change existing links.

You can use field mapping parameters in the following locations:

- Field mapping in the Mapplet transformation
- Field mapping in the Target transformation

mask rule

Represents a masking technique.

In the task, the mask rule parameter displays a list of masking techniques. You select and configure a masking technique in each incoming field.

Input parameter configuration

You can create a parameter in the Input Parameter panel or in the location where you want to use the parameter.

The Input Parameter panel displays all input parameters in the mapping. You can view details about the input parameter and the transformation where you use the parameter.

When you create a parameter in the Input Parameter panel, you can create any type of parameter. In a transformation, you can create the type of parameter that is appropriate for the location.

If you edit or delete an input parameter, consider how transformations that use the parameter might be affected by the change. For example, if a SQL transformation uses a connection parameter, the connection type must be **Oracle** or **SQL Server**. If the connection parameter is changed so that the connector type is no longer **Oracle** or **SQL Server**, the SQL transformation can no longer use the connection parameter.

To configure a mapping with a connection parameter, configure the mapping with a specific connection. Then, you can select the source, target, or lookup object that you want to use and configure the mapping. After the mapping is complete, you can replace the connection with a parameter without causing changes to other mapping details.

When you use an input parameter for a source, lookup, or target object, you cannot define the fields for the object in the mapping. Parameterize any conditions and field mappings in the data flow that would use fields from the parameterized object.

When you create an input parameter, you can use the parameter properties to provide guidance on how to configure the parameter in the task. The parameter description displays in the task as a tooltip, so you can add important information about the parameter value in the description.

The following table describes input parameter properties and how they display in a mapping task:

Input parameter property	Description
Name	Parameter name. Displays as the parameter name if you do not configure a display label. If you configure a display label, Name does not display in the task.
Display Label	Display label. Displays as the parameter name in the task.
Description	Description of the parameter. Displays as a tooltip for the parameter in the task. Use to provide additional information or instruction for parameter configuration.

Input parameter property	Description
Type	Parameter type. Determines where you can use the parameter. Also determines how the parameter displays in a mapping task: - String. Displays a textbox. For the Named Fields selection criteria, displays a list of fields. - Connection. Displays a list of connections. - Expression. Displays a Field Expression dialog box so you can create an expression. - Data object. Displays a list of available objects from the configured connection. - Field. Displays a list of fields from the selected object. - Field mapping. Displays field mapping tables allowing you to map fields from the data flow to the target object.
Connection Type	Determines the type of connection to use in the task. Applicable when the parameter type is Connection. For example, you select Oracle. Only Oracle connections are available in the task.
Allow parameter to be overridden at run time	Determines whether parameter values can be changed with a parameter file when the task runs. You define the parameter value to use in the task in the parameter file. When you configure the task, you specify a default value for the parameter. Applicable for data objects and connections with certain connection types. To see if a connector supports runtime override of source and target connections and objects, see the help for the appropriate connector. **Note:** If a mapping uses a source or target object parameter that can be overridden at runtime, and an existing object is selected in the task, the parameter value in the parameter file can't be null. If the value is null, the task fails
Default Value	Default value. Displays as the default value for the parameter, when available. For example, if you enter a connection name for a default value and the connection name does not exist in the organization, no default value displays.
Allow partial mapping override	Determines whether field mappings specified during mapping configuration can be changed in the task. Applicable when parameter type is Field mapping. Do not select **Allow Partial Mapping Override** if you want to enforce the links you configure in the mapping.

Partial parameterization with input parameters

To allow users to select one of the fields based on an input parameter at run time, you can implement partial parameterization in a mapping. Use partial parameterization to create templates for incremental data loads and other solutions.

For example, if you completely parameterize the source filter, you must include a query similar to the following example:

```
lastmodified_date > $$myvar
```

To partially parameterize the filter, you can specify the field as a variable, as shown in this example:

```
$field$ > $$myvar
```

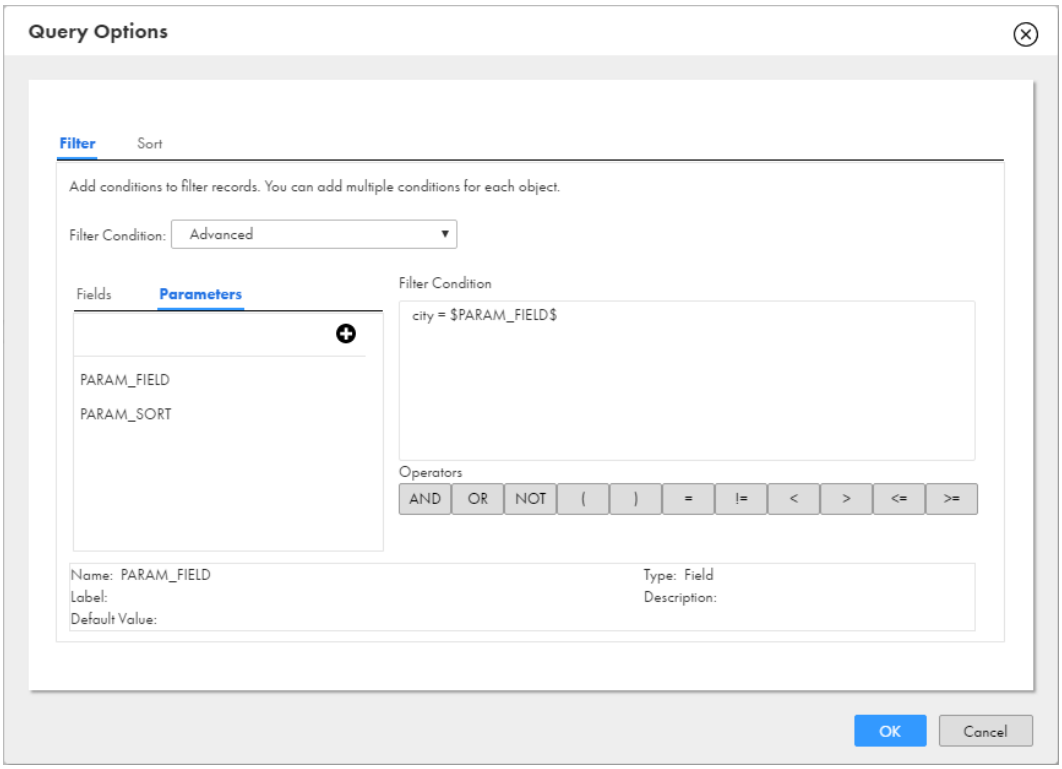
In this case, the user can select the required field in the mapping task.

To implement partial parameterization, you must use a database connection and a Source transformation advanced filter or a Filter, Expression, Router, or Aggregator transformation. You can create an input

parameter for one of the fields so that the user can select a specific field in the mapping task instead of writing a complete query. "String" and "field" are the only valid types.

Note: You can use the same parameter in all the supported transformations.

In the following example, the filter condition uses a parameter for the field name:



Rules and guidelines for partial parameterization

When you configure partial parameterization, note the following rules and guidelines:

- If you define a field type parameter in a Source transformation advanced filter, you can reuse it in a downstream transformation like a Router, Filter, Expression, or Aggregator. You cannot directly use field type parameters in other transformations.
- To distinguish parameters used for partial parameterization from in-out parameters (`$$myVar`), represent the parameter like an expression macro, for example, `$<Parameter_Name>$`.
- If you use a field type parameter in a Source transformation with multiple objects, qualify the parameter with the object name. You can either use the object name in the mapping or use a string type parameter to configure it in a mapping task.
- You cannot pass values for partial parameterization through a parameter file.
- You cannot use a user-defined function in an expression that uses partial parameterization. For example, the following expression is not valid:

  ```
  concat($Field$,:UDF.RemoveSpaces(NAME))
  ```

Using parameters in a mapping

When you use parameters in a mapping, you can change the parameter values each time you run the mapping task. You specify the parameter values in the mapping task or in a parameter file.

Use the following guidelines when you use parameters in a mapping:

When you create a mapping that includes source parameters, add the parameters after you configure the mapping.

For example, you have multiple customer account tables in different databases, and you want to run a monthly report to see customers for a specific state. When you create the mapping, you want to use parameters for the source connection, source object, and state. You update the parameter values to use at runtime when you configure the task.

To configure the mapping, you perform the following steps:

1. In the mapping, select the Source transformation.
2. On the **Source** tab, select a connection that contains one of the objects that you want to use, and then select the source object.
   You replace the source connection and object with parameters after the mapping is configured. You cannot select a source object if the source connection is a parameter. Add a source object so that you can configure the downstream data.
3. Add a Filter transformation.
4. On the **Filter** tab, add a filter condition. Select State for the field name, and create a new string parameter for the value. You resolve the parameter when you configure the task.
5. Configure the Target transformation.
6. Select the Source transformation.
7. On the **Source** tab, replace the source connection and the source object with parameters.

When you create a mapping with a parameterized target that you want to create at runtime, set the target field mapping to automatic.

If you create a mapping with a parameterized target object and you want to create the target at runtime, you must set the target field mapping to **Automatic** on the Target transformation **Field Mapping** tab. Automatic field mapping automatically links fields with the same name. You cannot map fields manually when you parameterize a target object.

In-out parameters

An in-out parameter is a placeholder for a value that stores a counter or task stage. Data Integration evaluates the parameter at run time based on your configuration.

In-out parameters act as persistent task variables. The parameter values are updated during task execution. The parameter might store a date value for the last record loaded from a data warehouse or help you manage the update process for a slowly changing dimension table.

For example, you might use an in-out parameter in one of the following ways:

Update values after each task execution.

You can use the SetVariable, SetMaxVariable, SetMinVariable, or SetCountVariable function in an Expression transformation to update parameter values each time you run a task.

To view the parameter values after the task completes, open the job details from the **All Jobs** or **My Jobs** page. You can also get these values when you work in the Mapping Designer or through the REST API.

Handle incremental data loading for a data warehouse.

In this case, you set a filter condition to select records from the source that meet the load criteria. When the task runs, you include an expression to increment the load process. You might choose to define the load process based on one of the following criteria:

- A range of records configured in an expression to capture the maximum value of the record ID to process in a session.
- A time interval, using parameters in an expression to capture the maximum date/time values, after which the session ends. You might want to evaluate and load transactions daily.

Parameterize an expression.

You might want to parameterize an expression and update it when the task runs. Create a string or text parameter and enable **Is expression variable**. Use the parameter in place of an expression and resolve the parameter at run time in a parameter file.

For example, you create the expression field parameter $$param and override the parameter value with the following values in a parameter file:

```
$$param=CONCAT(NAME,$$year)
$$year=2020
```

When the task runs, Data Integration concatenates the NAME field with 2020.

Parameterize a user-defined function.

Use an in-out parameter to parameterize all or part of a user-defined function in an expression.

Create a string parameter and enable **Is expression variable**. In the expression, you can use the in-out parameter as a placeholder for the user-defined function or for an argument in the user-defined function. Resolve the parameter at runtime in the mapping task or in the parameter file.

Note: Using in-out parameters in simultaneous mapping task runs can cause unexpected results.

You can use in-out parameters in the following transformations:

- Source
- Target
- Aggregator, but not in expression macros
- Expression, but not in expression macros
- Filter
- Router
- SQL
- Transaction Control

For each in-out parameter you configure the variable name, data type, default value, aggregation type, and retention policy. You can also use a parameter file that contains the value to be applied at run time. For a specific task run, you can change the value in the mapping task.

Unlike input parameters, an in-out parameter can change each time a task runs. The latest value of the parameter is displayed in the job details when the task completes successfully. The next time the task runs, the mapping task compares the in-out parameter to the saved value. You can also reset the in-out parameters in a mapping task, and then view the saved values in the job details.

In advanced mode, string and text in-out parameter values don't change each time the task runs. The mapping task always uses the same parameter value.

You can't use in-out parameters in mappings in SQL ELT mode.

Aggregation types

The aggregation type of an in-out parameter determines the final current value of the parameter when the task runs. You can use variable functions with a corresponding aggregation type to set the parameter value at run time.

You can select one of the following aggregation types for each parameter:

- Count
- Max
- Min

Variable functions

Variable functions determine how a task calculates the current value of an in-out parameter at run time.

You can use variable functions in an expression to set the current parameter value when a task runs.

To keep the parameter value consistent throughout the task run, use a valid aggregation type in the parameter definition. For example, you can use the SetMaxVariable function with the Max aggregation type but not the Min aggregation type.

The following table describes the available variable functions, aggregation types, and data types that you use with each function:

Variable function	Description	Valid aggregation type	Valid data type
SetVariable	Sets the parameter to the configured value. At the end of a task run, it compares the final current value to the start value. Based on the aggregation type, it saves a final value in the job details. This function is only available when the mapping runs on the Data Integration Server.	Max or Min	All transformation data types.
SetMaxVariable	Sets the parameter to the maximum value of a group of values. In advanced mode, this function is only available for the Expression transformation.	Max	All transformation data types except string and text data types are available in advanced mode.

Variable function	Description	Valid aggregation type	Valid data type
SetMinVariable	Sets the parameter to the minimum value of a group of values. In advanced mode, this function is only available for the Expression transformation.	Min	All transformation data types except string and text data types are available in advanced mode.
SetCountVariable	Increments the parameter value by one. In advanced mode, this function is only available for the Expression transformation. Configure the SetCountVariable function immediately before the target transformation to avoid a non-deterministic COUNT return value. For example, if you configure the SetCountVariable function before a transformation that contains multiple downstream pipelines, the generated COUNT value might be n times the actual row count.	Count	Integer and bigint

Note: Use variable functions one time for each in-out parameter in a pipeline. During run time, the task evaluates each function as it encounters the function in the mapping. As a result, the task might evaluate functions in a different order each time the task runs. This might cause inconsistent results if you use the same variable function multiple times in a mapping.

In-out parameter properties

Specify the parameter properties for each in-out parameter that you define.

The following table describes the in-out parameter properties:

In-out parameter property	Description
Name	Required. Name of the parameter. The parameter name cannot contain the text strings CurrentTaskName, CurrentTime, LastRunDate, or LastRunTime.
Description	Optional. Description that is displayed with the parameter in the job details and the mapping task. Maximum length is 255 characters.
Data Type	Required. Data type of the parameter. **Note:** Select a compatible aggregation type. For example, if you select string, you cannot configure it with the Count aggregation type.
Precision	Required. Precision of the parameter.
Scale	Optional. Scale of the parameter.

In-out parameter property	Description
Is expression variable	Optional. Controls whether Data Integration resolves the parameter value as an expression. Disable to resolve the parameter as a literal string. Applicable when the data type is String or Text. Default is disabled. Not applicable in advanced mode. Mappings in advanced mode resolve the parameter value as a literal string.
Default Value	Optional. Default value for the parameter, which might be the initial value when the mapping first runs. Use one of the following formats for datetime variables: - MM/DD/RR - MM/DD/YYYY - MM/DD/YYYY HH24:MI - MM/DD/RR HH24:MI - MM/DD/RR HH24:MI:SS - MM/DD/YYYY HH24:MI:SS - MM/DD/RR HH24:MI:SS.NS - MM/DD/YYYY HH24:MI:SS.NS
Retention Policy	Required. Determines when the mapping task retains the current value, based on the task completion status and the retention policy. Select one of the following options: - On success or warning (not applicable in advanced mode) - On success - On warning (not applicable in advanced mode) - Never
Aggregation Type	Required. Aggregation type of the variable. Determines the type of calculation you can perform and the available variable functions. Select one of the following options: - Count to count number of rows read from source. - Max to determine a maximum value from a group of values. - Min to determine a minimum value from a group of values.

In-out parameter values

An in-out parameter is a placeholder for a value or values that the task applies at run time. You define the value of the in-out parameter in the mapping and you can edit the value when you configure the mapping task.

A mapping task uses the following values to evaluate the in-out parameter at run time:

- Default Value. The value specified in the in-out parameter configuration.
- Value. The current value of the parameter as the task progresses. When a task starts, the value is the same as the default value. As the task progresses, the task calculates the value using a function that you set for the parameter. The task evaluates the value as each row passes through the mapping. Unlike the default value, the value can change. The task saves the final value in the job details after the task runs.

Note:

- If the task does not use a function to calculate the value of an in-out parameter, the task saves the default value of the parameter as the initial current value.
- An in-out parameter value cannot exceed 4000 characters.

At run time, the mapping task looks for the value in one of these locations, in the following order:

1. Value in the parameter file
2. Value saved from the previous task run
3. Default value in the mapping
4. Default value for the data type

If you want to override a saved value, define a value for the in-out parameter in a parameter file. The task uses the value in the parameter file.

Rules and guidelines for in-out parameters

Consider the following rules and guidelines:

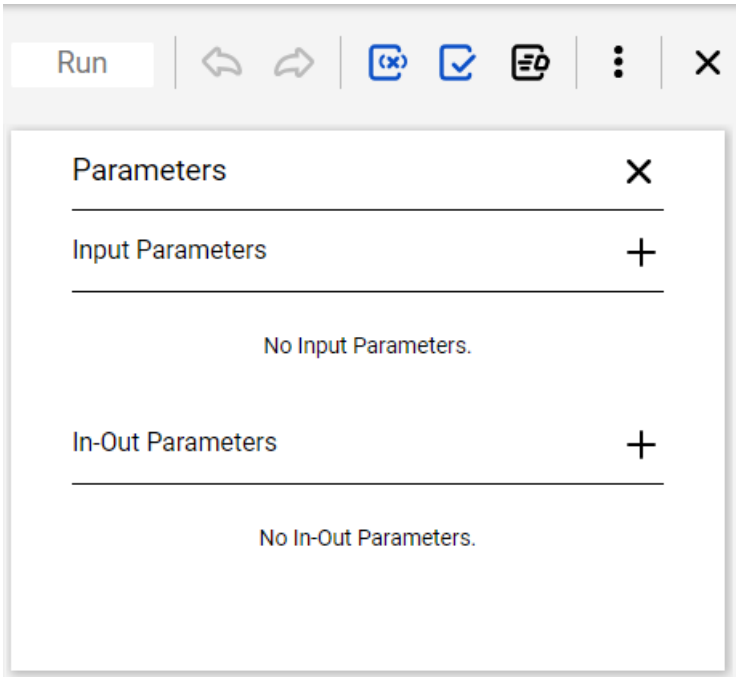
- When you write expressions that use in-out parameters, you don't need string identifiers for string variables.
- When you use a parameter in a transformation, enclose string parameters in string identifiers, such as single quotation marks, to indicate the parameter is a string.
- When you use in-out parameter in a source filter of type date/time, you must enclose the in-out parameter in single quotes because the value received after Informatica Intelligent Cloud Services resolves the in-out parameter can contain spaces.
- When you use an in-out parameter as a source query, the parameter must contain a valid query as the default value.
- If you copy, import, or export a mapping task, the session values of the in-out parameters are included.
- You can't use in-out parameters in a link rule or as part of a field name in a mapping.
- An in-out parameter name can't be the same as a field name or part of a field name.
- You can't use in-out parameters in an expression macro, because they rely on column names.
- When you use an in-out parameter in an expression or parameter file, precede the parameter name with two dollar signs ($$).
- For some connection types, when you use an in-out parameter for a date/time value, you cannot use $$ $SESSSTARTTIME to override the parameter value in a parameter file.
  For more information, see the help for the appropriate connector.
- An in-out parameter value can't exceed 4000 characters.
- A mapping task can't resolve a nested in-out parameter when the following conditions are true:
  - A parameter file updates an in-out parameter value in a mapping in advanced mode.
  - The parameter value is an expression that contains another in-out parameter.
- You can't preview data for sources and transformations with in-out parameters that are in mappings in advanced mode.

Creating an in-out parameter

You can configure an in-out parameter from the Mapping Designer or the Mapplet Designer.

1. In the Mapping Designer or Mapplet Designer, add the transformation where you want to use an in-out parameter and add the upstream transformations.
2. Open the Parameters panel.

The In-Out Parameters display beneath the Input Parameters.



3. Add an in-out parameter.
4. Configure the parameter properties.
5. Use the parameter as a variable in the transformation where you want to set the value when the mapping runs.

For details on the in-out parameter properties and the Parameters panel, see "In-out parameter properties" on page 73 and "Mapping Designer" on page 10.

Editing in-out parameters in a mapping task

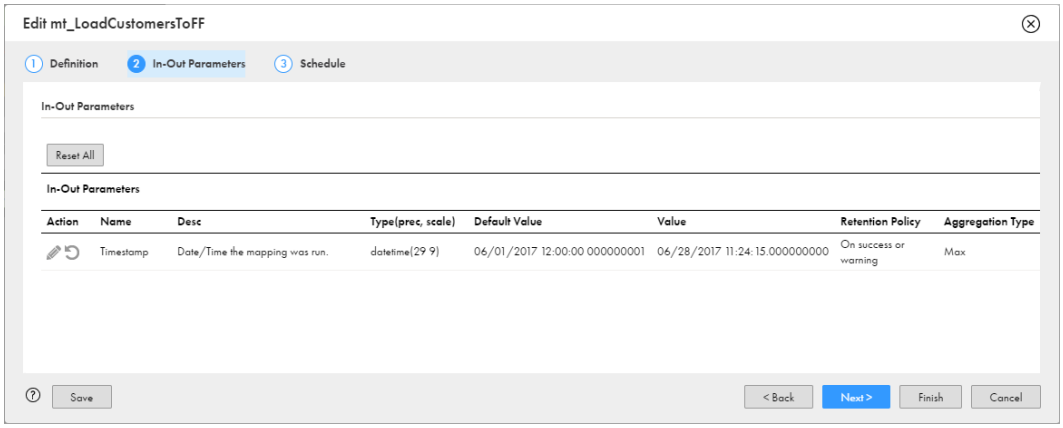
An in-out parameter is a placeholder for a value in a mapping. The task determines the value to apply during run time. You configure an in-out parameter in the mapping and can edit the value in the mapping task.

When you deploy a mapping that includes an in-out parameter, the task sets the parameter value at run time based on the parameter's retention policy. By default, the mapping task retains the value set during the last session. If needed, you can reset the value in the mapping task.

From the mapping task wizard, you can perform the following actions for in-out parameters:

- View the values of all in-out parameters in the mapping, which can change each time the task runs.
- Reset the configuration to the default values. Click **Refresh** to reset a single parameter. Click **Refresh All** to reset all the parameters.
- Edit or change specific configuration details. Click **Edit**.

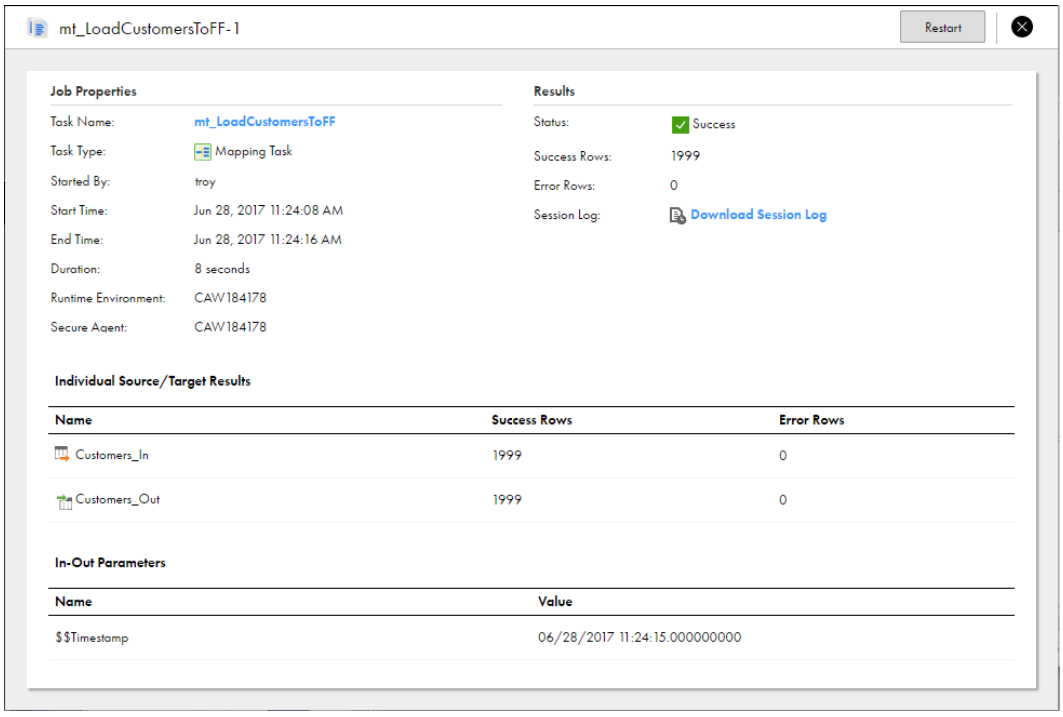
For example, the following image shows configuration details of the "Timestamp" parameter and the value at the end of the last session:



View in-out parameters in the job details

To find the value of an in-out parameter after a task runs, view the job details. To view job details, open Monitor and select **All Jobs** or open Data Integration and select **My Jobs**. Then click the job name.

The following image shows an example of the available details, including the current value of the specified parameter, set during the last run of a mapping task:



The in-out parameters appear in the job details based on the retention policy that you set for each parameter.

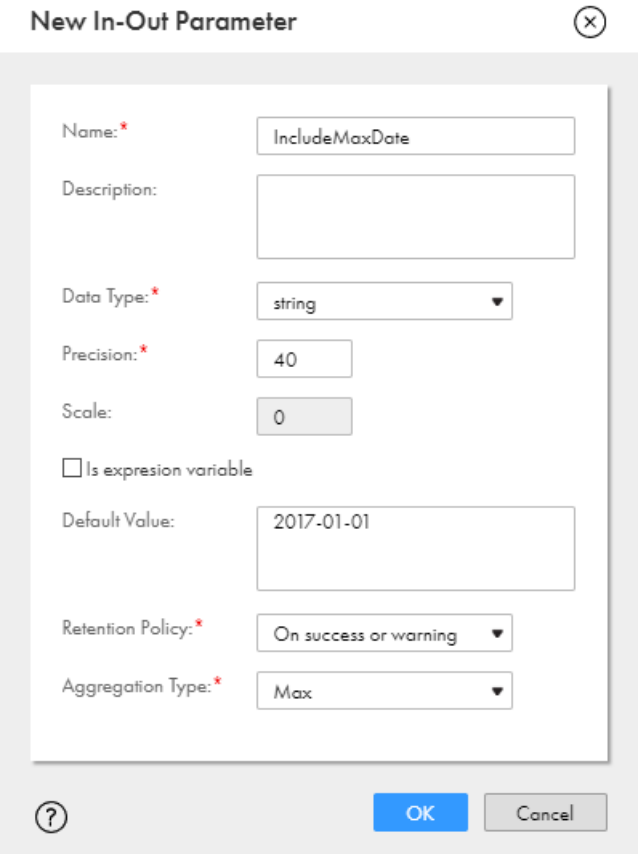
In-out parameter example

You can use an in-out parameter as a persistent task variable to manage an incremental data load.

The following example uses an in-out parameter to set a date counter for the task and perform an incremental read of the source. Instead of manually entering a task override to filter source data each time the task runs, the mapping contains a parameter, $$IncludeMaxDate.

In the example shown here, the in-out parameter is a date field where you want to support the MM/DD/YYYY format. To support this format, you can use the SetVariable function in the Expression transformation and a string data type.

In the Mapping Designer, you open the **Parameters** panel and configure an in-out parameter as shown in the following image:



The sample mapping has the following transformations:

- The **Source transformation** applies the following filter to select rows from the users table where the transaction date, TIMESTAMP, is greater than the in-out parameter, $$IncludeMaxDate:

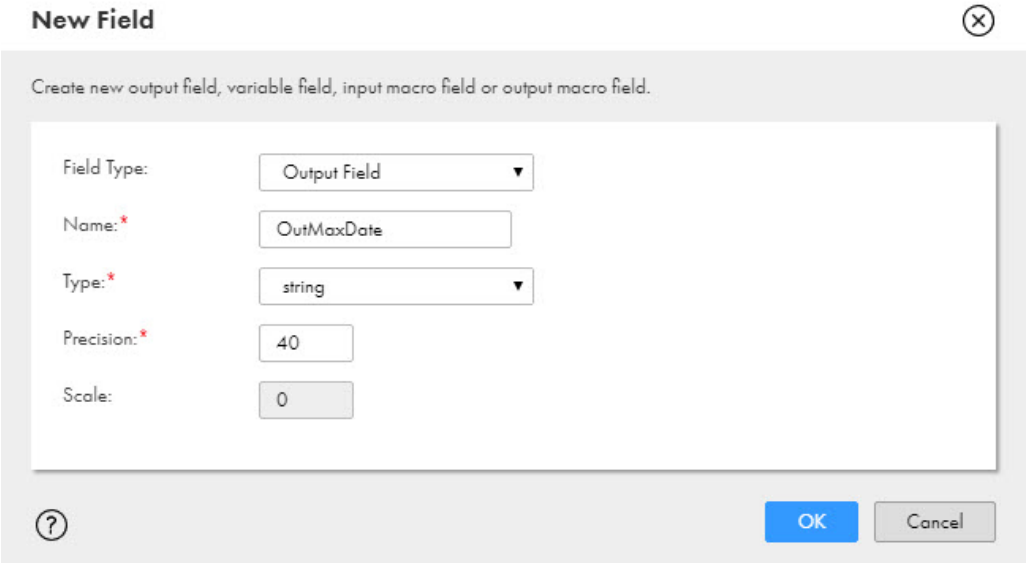
  ```
  users.TIMESTAMP > '$$IncludeMaxDate'
  ```

  The Source transformation also applies the following sort order to the output to simplify the expression in the next transformation:

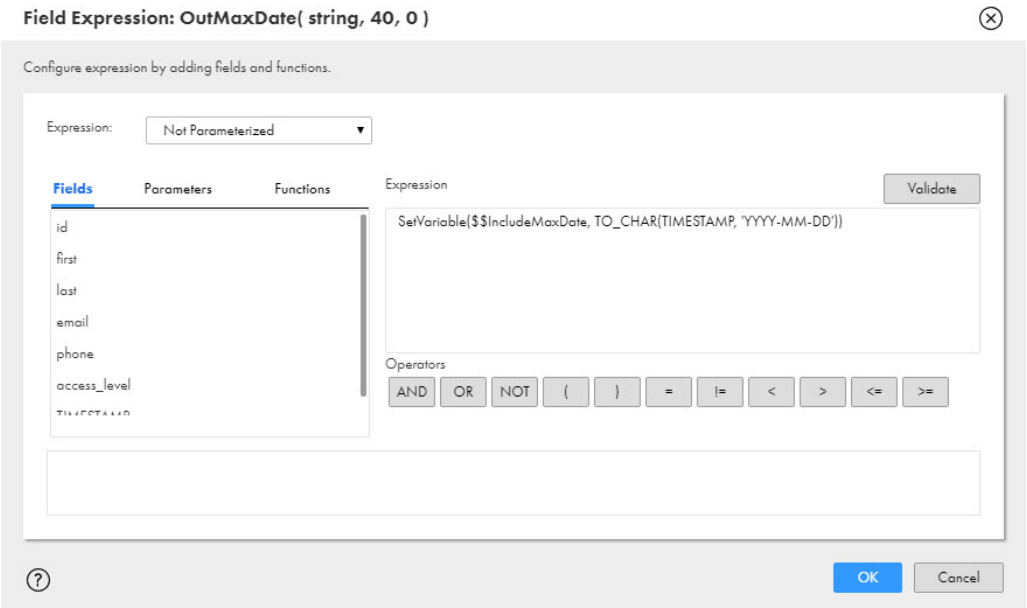
  ```
  users.TIMESTAMP (Ascending)
  ```

- The **Expression transformation** contains a simple expression that sets the current value of $$IncludeMaxDate.

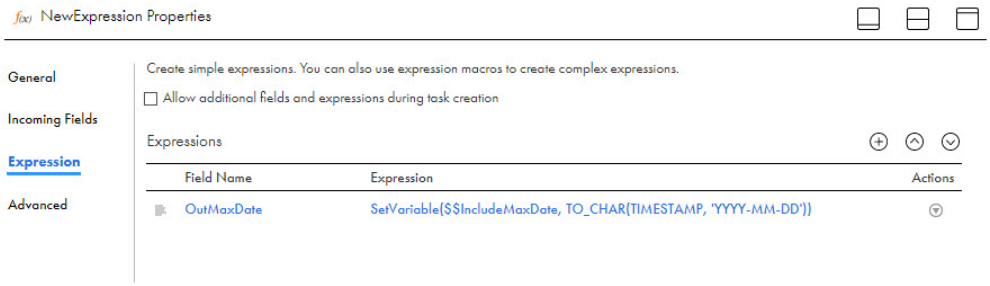
The Expression output field, OutMaxDate, is a string type that enables you to map the expression output to the target.



The SetVariable function sets the current parameter value each time the session runs. For example, if you set the default value of $$IncludeMaxDate to 2016-04-04, the task reads rows dated through 2016-04-04 the first time it runs. The task sets $$IncludeMaxDate to 2016-04-04 when the session is complete. The next time the session runs, the task reads rows with a date greater than 2016-04-04 based on the source filter.



You can view the saved expression for OutMaxDate, which also converts the source column to a DATE_ID in the format YYYY-MM-DD.



- The **Target transformation** maps the Expression output field to a target column.

When the mapping runs, the OutMaxDate contains the last date for which the task loaded records.

In-out parameter example in advanced mode

You can use an in-out parameter as a persistent task variable to manage an incremental data load.

The following example uses an in-out parameter to set a date counter for the task and perform an incremental read of the source. Instead of manually entering a task override to filter source data each time the task runs, the mapping contains a parameter, $$IncludeMaxDate. This example is based on a relational database source with an incremental timestamp column.

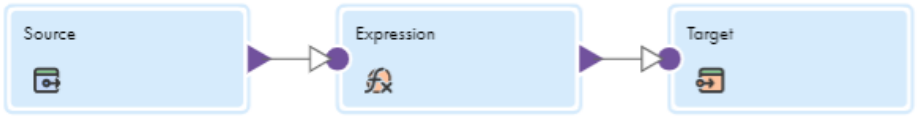
The high level steps is this example include:

1. Create a mapping.
2. Create and define the in-out parameter.
3. Configure the filter condition and source in the Source transformation.
4. Add an Expression transformation and configure the SetMaxVariable function.

Create a mapping

Mappings contain the Source transformation and Target transformation by default.

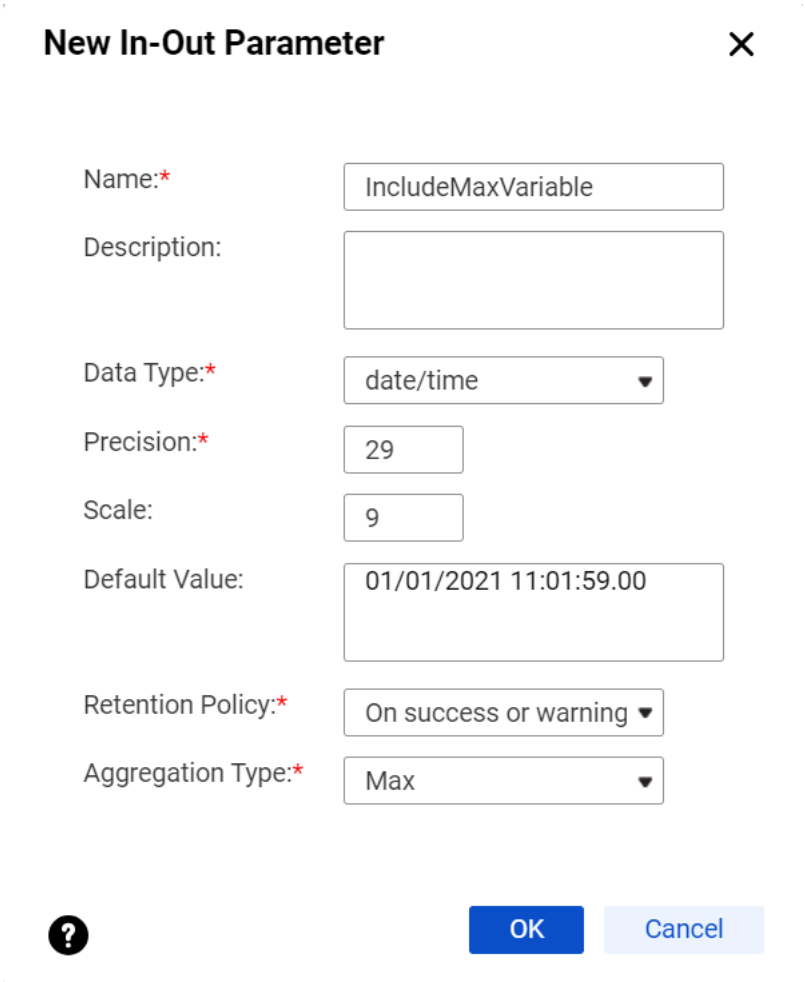
The following image shows a fully configured mapping.



Create and define the in-out parameter

The in-out parameter is a date field where you want to use the MM/DD/YYYY HH24:MI:SS.NS format.

In the Mapping Designer, open the **Parameters** panel and configure an in-out parameter as shown in the following image:



Configure the filter condition and source in the Source transformation

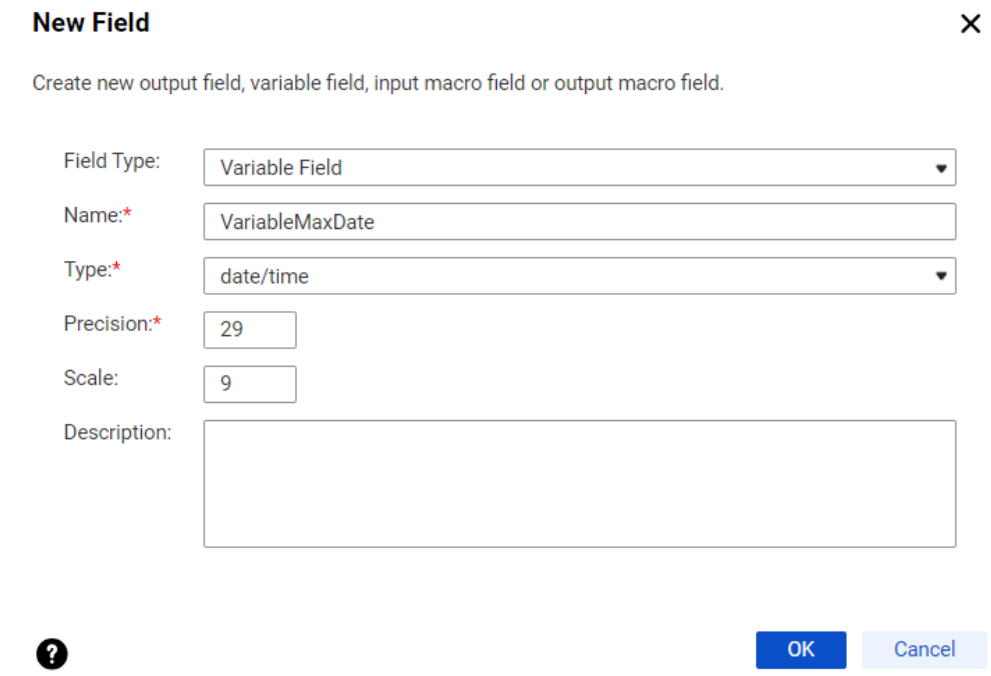
Use the Source filtering options in the Source transformation to apply the following filter to select rows from the users table where the transaction date, TIMESTAMP, is greater than the in-out parameter, $$IncludeMaxDate:

```
users.TIMESTAMP > '$$IncludeMaxDate'
```

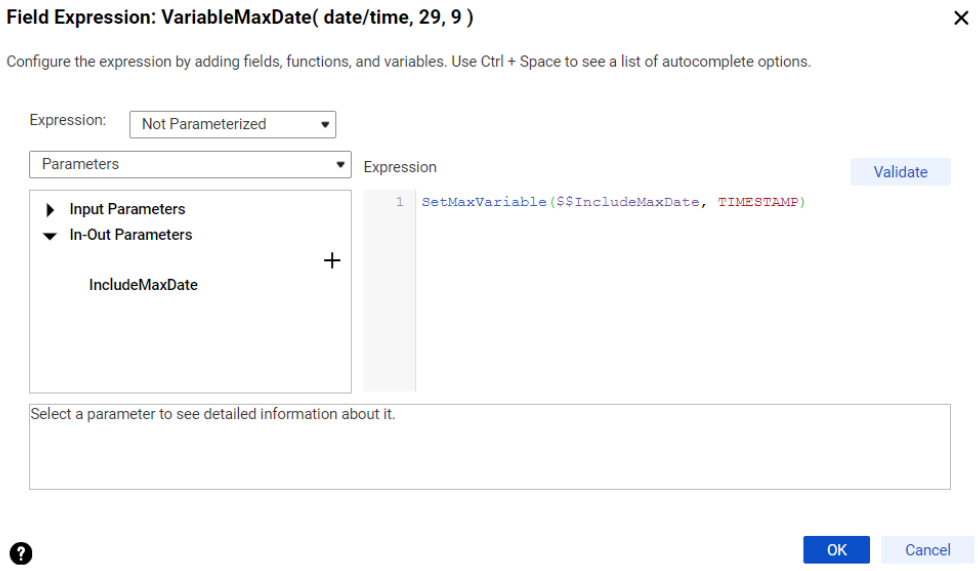
Add an Expression transformation and configure the SetMaxVariable function

The Expression transformation contains a simple expression that sets the current value of $$IncludeMaxDate.

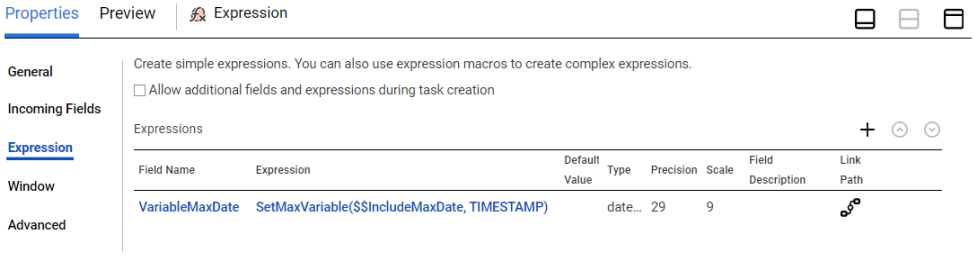
The New Field dialog box shows the **Field Type** as Variable Field, **Name** as VariableMaxDate, **Type** as date/time, and **Precision** as 29.



The SetMaxVariable function sets the current parameter value each time the task runs. For example, if you set the default value of $$IncludeMaxDate to 01/01/2021 11:01:59.00, the task reads rows dated through January 1, 2021 the first time it runs. For the first task run, you specify the start date based on your needs. The task sets $$IncludeMaxDate to 01/11/2021 10:00:00.00 when the session is complete. The next time the task runs, it reads rows with a date/time value greater than 01/11/2021 10:00:00.00 based on your configuration of the Source filtering options.



You can view the saved expression for VariableMaxDate.



After the mapping runs successfully, the in-out parameter contains the last date for which the task loaded data.

Using in-out parameters as expression variables

Configure an in-out parameter as an expression variable and use it to parameterize expressions or user-defined functions. To use an in-out parameter as an expression variable, create a string parameter and enable the **Is expression variable** option. You can't use an in-out parameter as an expression variable in advanced mode.

When you enable **Is expression variable**, Data Integration resolves the parameter as an expression. When you disable this option, Data Integration resolves the parameter as a literal string.

You can use an in-out parameter as an expression variable in the following transformations:

- Aggregator
- Expression
- Filter
- Router

You can override the parameter at runtime with a value specified in a parameter file.

1. In the mapping, create an in-out parameter.
2. Configure the parameter properties.
3. Set the data type to String or Text.
4. Enable the **Is expression variable** option.
5. Use the parameter as an expression or user-defined function.
6. Optionally, you can override the default value of the parameter in one of the following places:
   - On the **In-Out Parameters** tab of the task.
   - In a parameter file. Enter the parameter file name and location on the **Schedule** tab of the task.

When the task runs, Data Integration resolves the parameter as an expression.

Using in-out parameters as source queries

You can use an in-out parameter as a placeholder for a source query. To use an in-out parameter as a source query, create a string parameter, define a default query value, and enter the parameter in the query dialog of the Source transformation properties.

When you use an in-out parameter as a source query, you must provide a valid query for the default parameter value. Data Integration uses the default query to generate design time source metadata. Data Integration uses this metadata to map fields to the fields in the query that you use at run time. If Data Integration can't generate the design time metadata, the task fails.

You can create the in-out parameter before or after you configure the Source transformation. However, if you configure the Source transformation before you define the default parameter value, you must synchronize the source fields on the **Fields** tab before you run the task.

1. In the Mapping Designer, create an in-out parameter and enter a default query.
2. Configure the Source transformation.
3. Select **Query** as the source type and enter the in-out parameter in the **Query** dialog using the following format:

   ```
   $$<parameter name>
   ```
4. In the mapping task, perform one of the following actions:
   - On the **Source** tab, enter the query that you want to use at run time.
   - On the **Schedule** or **Runtime Options** tab, enter the parameter file details.

Parameter files

A parameter file is a list of user-defined parameters and their associated values.

Use a parameter file to define values that you want to update without having to edit the task. You update the values in the parameter file instead of updating values in a task. The parameter values are applied when the task runs.

You can use a parameter file to define parameter values in the following tasks:

Mapping tasks

Define parameter values for connections in the following transformations:

- Source
- Target
- Lookup
- SQL

Define parameter values for objects in the following transformations:

- Source
- Target
- Lookup

Also, define values for parameters in data filters, expressions, and lookup expressions.

Note: Not all connectors support parameter files. To see if a connector supports runtime override of connections and data objects, see the help for the appropriate connector.

Synchronization tasks

Define values for parameters in data filters, expressions, and lookup expressions.

PowerCenter tasks

Define values for parameters and variables in data filters, expressions, and lookup expressions.

You enter the parameter file name and location when you configure the task.

You can't use a parameter file in a mapping task that is based on a mapping in SQL ELT mode.

Parameter file requirements

You can reuse parameter files across assets such as mapping tasks, taskflows, and linear taskflows. To reuse a parameter file, define local and global parameters within a parameter file.

You group parameters in different sections of the parameter file. Each section is preceded by a heading that identifies the project, folder, and asset to which you want to apply the parameter values. You define parameters directly below the heading, entering each parameter on a new line.

The following table describes the headings that define each section in the parameter file and the scope of the parameters that you define in each section:

Heading	Description
#USE_SECTIONS	Tells Data Integration that the parameter file contains asset-specific parameters. Use this heading as the first line of a parameter file that contains sections. Otherwise Data Integration reads only the first global section and ignores all other sections.
[Global]	Defines parameters for all projects, folders, tasks, taskflows, and linear taskflows.
[project name].[folder name].[taskflow name] -or- [project name].[taskflow name]	Defines parameters for tasks in the named taskflow only. If a parameter is defined in a taskflow section and in a global section, the value in the taskflow section overrides the global value.
[project name].[folder name].[linear taskflow name] -or- [project name].[linear taskflow name]	Defines parameters for tasks in the named linear taskflow only. If a parameter is defined in a linear taskflow section and in a global section, the value in the linear taskflow section overrides the global value.
[project name].[folder name].[task name] -or- [project name].[task name]	Defines parameters for the named task only. If a parameter is defined in a task section and in a global section, the value in the task section overrides the global value. If a parameter is defined in a task section and in a taskflow or linear taskflow section and the taskflow uses the task, the value in the task section overrides the value in the taskflow section.

If the parameter file does not contain sections, Data Integration reads all parameters as global.

Precede the parameter name with two dollar signs, as follows: $$<parameter>. Define parameter values as follows:

```
$$<parameter>=value
$$<parameter2>=value2
```

For example, you have the parameters SalesQuota and Region. In the parameter file, you define each parameter in the following format:

```
$$SalesQuota=1000
$$Region=NW
```

The parameter value includes any characters after the equals sign (=), including leading or trailing spaces. Parameter values are treated as String values.

Parameter scope

When you define values for the same parameter in multiple sections in a parameter file, the parameter with the smallest scope takes precedence over parameters with larger scope.

In this case, Data Integration gives precedence to parameter values in the following order:

1. Values defined in a task section.
2. Values defined in a taskflow or linear taskflow section.
3. Values defined in the #USE_SECTIONS section.
4. Values defined in a global section.

For example, a parameter file contains the following parameter values:

```
[GLOBAL]
$$connection=ff5
[Project1].[Folder1].[monthly_sales]
$$connection=ff_jd
```

For the task "monthly_sales" in Folder1 inside Project1, the value for parameter $$connection is "ff_jd." In all other tasks, the value for $$connection is "ff5."

If you define a parameter in a task section and in a taskflow or linear taskflow section and the taskflow uses the task, Data Integration uses the parameter value defined in the task section.

For example, you define the following parameter values in a parameter file:

```
#USE_SECTIONS
$$source=customer_table
[GLOBAL]
$$location=USA
$$sourceConnection=Oracle
[Default].[Sales].[Task1]
$$source=Leads_table
[Default].[Sales].[Taskflow2]
$$source=Revenue
$$sourceconnection=ODBC_1
[Default].[Taskflow3]
$$source=Revenue
$$sourceconnection=Oracle_DB
```

Task1 contains the $$location, $$source, and $$sourceconnection parameters. Taskflow2 and Taskflow3 contain Task1.

When you run Taskflow2, Data Integration uses the following parameter values:

Parameter	Section	Value
$$source	[Default].[Sales].[Task1]	Leads_table
$$sourceconnection	[Default].[Sales].[Taskflow2]	ODBC_1
$$location	[GLOBAL]	USA

When you run Taskflow3, Data Integration uses the following parameter values:

Parameter	Section	Value
$$source	[Default].[Sales].[Task1]	Leads_table
$$sourceconnection	[Default].[Taskflow3]	Oracle_DB
$$location	[GLOBAL]	USA

When you run Task1, Data Integration uses the following parameter values:

Parameter	Section	Value
$$source	[Default].[Sales].[Task1]	Leads_table
$$sourceconnection	[GLOBAL]	Oracle
$$location	[GLOBAL]	USA

For all other tasks that contain the $$source parameter, Data Integration uses the value customer_table.

Sample parameter file

The following example shows a sample parameter file entry:

```
#USE_SECTIONS
$$oracleConn=Oracle_SK
$$city=SF
[Global]
$$ff_conn=FF_ja_con
$$st=CA
[Default].[Accounts].[April]
$$QParam=SELECT * from con.ACCOUNT where city=LAX
$$city=LAX
$$tarOb=accounts.csv
$$oracleConn=Oracle_Src
```

Parameter file location

When you use a parameter file, save the parameter file on a local machine or in a cloud-hosted directory based on the task type. You enter details about the parameter file on the **Schedule** tab when you create the task.

The following table lists the default parameter file directory for each task type:

Task type	Default parameter file directory
Mapping task based on a mapping	`<Secure Agent installation directory>/apps/Data_Integration_Server/data/userparameters`
Mapping task based on a mapping in advanced mode	`<Secure Agent installation directory>/apps/data/userparameters`
Synchronization task	`<Secure Agent installation directory>/apps/Data_Integration_Server/data/userparameters`

For mapping tasks, you can also save the parameter file in one of the following locations:

A local machine

Save the file in a location that the Secure Agent can access. For mappings in advanced mode, the file must be stored on the Secure Agent machine.

You enter the file name and directory on the **Schedule** tab when you create the task. Enter the absolute file path. Alternatively, enter a path relative to a $PM system variable, for example, `$PMRootDir/ParameterFiles`.

The following table lists the system variables that you can use:

System variable	Description
$PMRootDir	Root directory for the Data Integration Server Secure Agent service. Default is `<Secure Agent installation directory>/apps/Data_Integration_Server/data`.
$PMBadFileDir	Directory for row error logs and reject files. Default is $PMRootDir/error.
$PMCacheDir	Directory for index and data cache files. Default is $PMRootDir/cache.
$PMExtProcDir	Directory for external procedures. Default is $PMRootDir.
$PMLookupFileDir	Directory for lookup files. Default is $PMRootDir.
$PMSessionLogDir	Directory for session logs. Default is $PMRootDir/../logs.
$PMSourceFileDir	Directory for source files. Default is $PMRootDir.
$PMStorageDir	Directory for files related to the state of operation of internal processes such as session and workflow recovery files. Default is $PMRootDir.

System variable	Description
$PMTargetFileDir	Directory for target files. Default is $PMRootDir.
$PMTempDir	Directory for temporary files. Default is $PMRootDir/temp.
$PMWorkflowLogDir	Directory for workflow logs. Default is $PMRootDir/../logs.

To find the configured path of a system variable, see the pmrdtm.cfg file located at the following directory:

```
<Secure Agent installation directory>\apps\Data_Integration_Server\<Data Integration Server version>\ICS\main\bin\rdtm
```

You can also find the configured path of any variable except $PMRootDir in the Data Integration Server system configuration details in Administrator.

If you do not enter a location, Data Integration uses the default parameter file directory.

A cloud platform

You can use a connection stored with Informatica Intelligent Cloud Services. The following table shows the connection types that you can use and the configuration requirements for each connection type:

Connection type	Requirements
Amazon S3 V2	You can use a connection that was created with the following credentials: - Access Key - Secret Key - Region The S3 bucket must be public.
Azure Data Lake Store Gen2	You can use a connection that was created with the following credentials: - Account Name - ClientID - Client Secret - Tenant ID - File System Name - Directory Path The storage point must be public.
Google Storage V2	You can use a connection that was created with the following credentials: - Service Account ID - Service Account Key - Project ID The storage bucket must be public.

Create the connection before you configure the task. You select the connection and file object to use on the **Schedule** tab when you create the task.

Data Integration displays the location of the parameter file and the value of each parameter in the job details after you run the task.

Rules and guidelines for parameter files

Data Integration uses the following rules to process parameter files:

- If a parameter isn't defined in the parameter file, Data Integration uses the value defined in the task.
- If a mapping uses a source or target object parameter that can be overridden at runtime and an existing object is selected in the task, the parameter value in the parameter file can't be null. If the value is null, the task fails.
- In advanced mode, you can't set input parameter values in a parameter file. If you override an in-out parameter in a parameter file that relies on an input parameter value, make sure the in-out parameter value is valid based on the input parameter values that you configure in the mapping task. Otherwise, the mapping might fail.

  For example, if the source connection is an input parameter and the source table is an in-out parameter, the table name that you configure in the parameter file must exist in the source connection that you set in the mapping task.
- In advanced mode, make sure that default parameter values are valid for schema and table names. If you use a parameter file to update the parameter value for a schema name or table name override and the default parameter value is invalid, the mapping fails.
- Data Integration processes the file top-down.
- If a parameter value is defined more than once in the same section, Data Integration uses the first value. For example, a parameter file contains the following task section:

  ```
  [MyProject].[Folder1].[mapping_task1]
  $$sourceconn=Oracle
  $$filtervariable=ID
  $$sourceObject=customer_table
  $$targetconn=salesforce
  $$sourceconn=ff_2
  ```

  When mapping_task1 runs, the value of the sourceconn parameter is Oracle.
- If a parameter value is another parameter defined in the file, precede the parameter name with one dollar sign ($). Data Integration uses the first value of the variable in the most specific scope. For example, a parameter file contains the following parameter values:

  ```
  [GLOBAL]
  $$ffconnection=my_ff_conn
  $$var2=California
  $var5=North
  [Default].[folder5].[sales_accounts]
  $$var2=$var5
  $var5=south
  ```

  In the task "sales_accounts," the value of "var5" is "south." Since var2 is defined as var5, var2 is also "south."

  In advanced mode, precede the parameter name with two dollar signs ($$):

  ```
  [GLOBAL]
  $$ffconnection=my_ff_conn
  $$var2=California
  $$var5=North
  [Default].[folder5].[sales_accounts]
  $$var2=$$var5
  $$var5=south
  ```
- If a task is defined more than once, Data Integration combines the sections.
- If a parameter is defined in multiple sections for the same task, Data Integration uses the first value. For example, a parameter file contains the following task sections:

  ```
  [Default].[Folder1].[MapTask2]
  $$sourceparam=Oracle_Cust
  ```

```
[Default].[Folder1].[MapTask2]
$$sourceparam=Cust_table
$$targetparam=Sales
```

When you run MapTask2, Data Integration uses the following parameter values:

- $$sourceparam=Oracle_Cust
- $$targetparam=Sales

- The value of a parameter is global unless it is present in a section.
- Data Integration ignores sections with syntax errors.
- The default date/time format for a mapping in advanced mode is MM/DD/YYYY HH24:MI:SS. To use a different date/time format, configure the advanced session property `DateTime Format String` in the mapping task.

Parameter file templates

You can generate and download a parameter file template that contains mapping parameters and their default values. The parameter file template includes input and in-out parameters that can be overridden at runtime. Save the parameter file template and use it to apply parameter values when you run the task, or copy the mapping parameters to another parameter file.

When you generate a parameter file template, the file contains the default parameter values from the mapping on which the task is based. If you do not specify a default value when you create the parameter, the value for the parameter in the template is blank.

The parameter file template does not contain the following elements:

- Runtime values that you specify in the mapping task
- Partial parameters
- Advanced session properties
- System defined variables

If you add, edit, or delete parameters in the mapping, download a new parameter file template.

Downloading a parameter file template

1. On the **Schedule** tab in the mapping task, click **Download Parameter File Template**.

   The file name is `<mapping task name>.param`.
2. If you want to use the file in subsequent task runs, save the parameter file in a location that is accessible by the Secure Agent.

   Enter the file name and directory on the **Schedule** tab when you configure the task.

Overriding connections with parameter files

If you use a connection parameter in a mapping, you can override the connection defined in the mapping task at runtime with values specified in a parameter file.

When you define a connection value in a parameter file, the connection type must be the same as the default connection type in the mapping task. For example, you create a Flat File connection parameter and use it as the source connection in a mapping. In the mapping task, you provide a flat file default connection. In the parameter file, you can only override the connection with another flat file connection.

When you override an FTP connection using a parameter, the file local directory must the same.

You cannot use a parameter file to override a lookup with an FTP/SFTP connection.

Note: Some connectors support only cached lookups. To see which type of lookup a connector supports, see the help for the appropriate connector.

1. In the mapping, create an input parameter:
   a. Select **connection** for the parameter type .
   b. Select **Allow parameter to be overridden at runtime**.
2. In the mapping, use the parameter as the connection that you want to override.
3. In the mapping task, define the parameter details:
   a. Select a default connection.
   b. On the **Schedule** tab, enter the parameter file directory and parameter file name.
4. In the parameter file, define the connection parameter with the value that you want to use at runtime.

   Precede the parameter name with two dollar signs ($$). For example, you have a parameter with the name ConParam and you want to override it with the connection OracleCon1. You define the runtime value with the following format:

   ```
   $$ConParam=OracleCon1
   ```
5. If you want to change the connection, update the parameter value in the parameter file.

Overriding data objects with parameter files

If you use a data object parameter in a mapping, you can override the object defined in the mapping task at runtime with values specified in a parameter file.

Note: You cannot override source objects when you read from multiple relational objects or from a file list. You cannot override target objects if you create a target at run time.

When you define an object parameter in the parameter file, the parameter in the file must have the same metadata as the default parameter in the mapping task. For example, if you override the source object ACCOUNT with EMEA_ACCOUNT, both objects must contain the same fields and the same data types for each field.

1. In the mapping, create an input parameter:
   a. Select **data object** for the parameter type.
   b. Select **Allow parameter to be overridden at runtime**.
2. In the mapping, use the object parameter at the object that you want to override.
3. In the mapping task, define the parameter details:
   a. Set the type to **Single**.
   b. Select a default data object.
   c. On the **Schedule** tab, enter the parameter file directory and file name.
4. In the parameter file, specify the object to use at runtime. Perform one of the following tasks:
   - For sources other than flat file or FTP sources, precede the parameter name with two dollar signs ($$). For example, you have a parameter with the name ObjParam1 and you want to override it with the data object SourceTable. You define the runtime value with the following format:

     ```
     $$ObjParam1=SourceTable
     ```

- For flat file and FTP sources, precede the parameter name with one dollar sign ($). The following example is from a parameter file that's used to override the source directory, source file name, and remote file name for an FTP connection:

```
#USE_SECTIONS
 [MyProjectFolder].[MappingTask_ftp]
$Source_dir=/data/test/test_data/Test/
$Source_file=src_ff.txt
$RemoteFileName=/root/FlatFile/Scotty/Test/src_14892_1.txt
 [Global]
```

5. If you want to change the object, update the parameter value in the parameter file.

Overriding source queries

If you use a source query or filter condition in a mapping, you can override the value specified in the mapping task with values specified in a parameter file. You can override source queries for relational and ODBC database connections.

When you define an SQL query, the fields in the overridden query must be the same as the fields in the default query. The task fails if the query in the parameter file contains fewer fields or is invalid.

If a filter condition parameter is not resolved in the parameter file, Data Integration will use the parameter as the filter value and the task returns zero rows.

1. In the mapping, create a data object parameter.
2. Select **Allow parameter to be overridden at runtime**.
3. Use the parameter as the source object.
4. In the mapping task, on the **Sources** tab, select **Query** as the source type.
5. Enter a default custom query.
6. On the **Schedule** tab, provide the parameter file name and location.
7. In the parameter file, enter the values to use when the task runs.
8. If you want to change the query, update the parameter value in the parameter file.

Creating target objects at run time with parameter files

If you use a target object parameter in a mapping, you can create a target at run time using a parameter file.

You include the target object parameter and the name that you want to use in a parameter file. If the target name in the parameter file doesn't exist, Data Integration creates the target at run time. In subsequent runs, Data Integration uses the existing target.

To create a target at run time using a parameter file, the following conditions must be true:

- The mapping uses a flat file, relational, or file storage-based connection.
- The mapping is used in a mapping task, dynamic mapping task, or taskflow.

For file storage-based connections, the parameter value in the parameter file can include both the path and file name. If the path is not specified, the target is created in the default path specified in the connection.

1. In the mapping, create an input parameter:
   a. Select **data object** for the parameter type.
   b. Select **Allow parameter to be overridden at runtime**.
2. In the mapping, use the parameter as the target object.

3. In the task, define the parameter details:
   a. Set the type to **Single**.
   b. Select a default data object.
   c. On the **Schedule** tab, enter the parameter file directory and file name.
4. In the parameter file, specify the name of the target object that you want to create.

   Precede the parameter name with two dollar signs ($$). For example, you have a parameter with the name TargetObjParam and you want to create a target object with the name MyTarget. You define the runtime value with the following format:

   ```
   $$TargetObjParam=MyTarget
   ```

   For file storage-based connector types, include the path in the object name. If you don't include a path, the target is created in the default path specified in the connection.
5. If you want to change the object, update the parameter value in the parameter file.

CHAPTER 4

CLAIRE recommendations

If your organization has enabled CLAIRE recommendations, you can receive recommendations during mapping design. CLAIRE, Informatica's AI engine, uses machine learning to make recommendations based on the current flow of the mapping and metadata from prior mappings across Informatica Intelligent Cloud Services organizations.

When your organization opts in to receive CLAIRE-based recommendations, anonymous metadata from your organization's mappings is analyzed and leveraged to offer design recommendations.

To disable recommendations for the current mapping, use the recommendation toggle. You can enable recommendations again at any time.

When you create a new mapping, recommendations are enabled by default. If you edit an existing mapping, recommendations are disabled by default.

CLAIRE can make the following types of recommendations during mapping design:

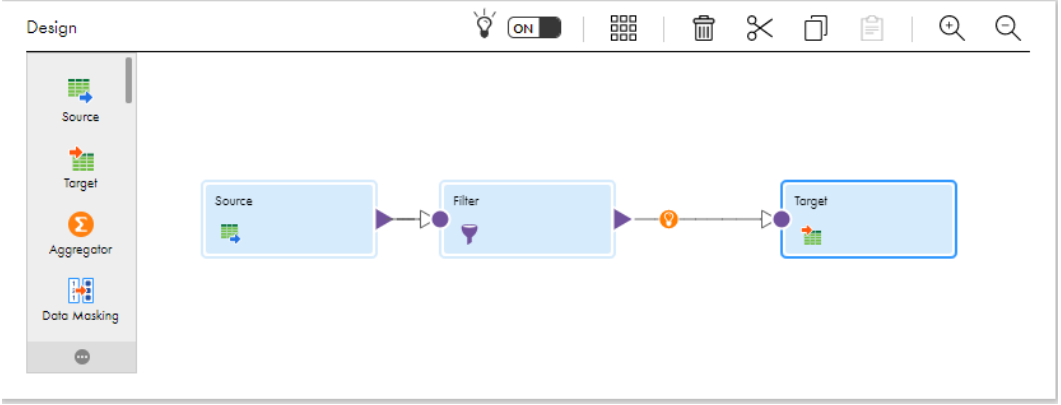
- Recommend transformation types to include in the mapping.
- Recommend additional sources based on primary key and foreign key relationships.
- Recommend joining or unioning additional sources.
- Make recommendations for objects in a mapping inventory.
- Recommend mapplets or user-defined functions.
- Make recommendations to standardize your data
- Make recommendations to mask sensitive data.
- Recommend rule specifications.

CLAIRE recommendations aren't available for mappings in SQL ELT mode.

Transformation type recommendations

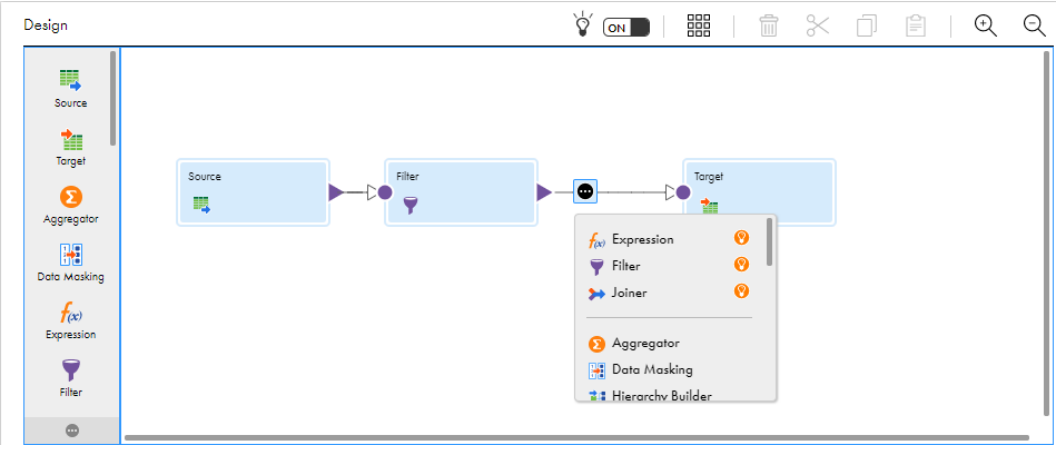
CLAIRE uses design metadata and the current flow of your mapping to recommend transformations in the data flow. CLAIRE polls the mapping after every change to provide the most relevant recommendations.

When CLAIRE detects a transformation to add, the Add Transformation icon displays orange on the transformation link as shown in the following image:



Click the Add Transformation icon to display the Add Transformation menu. Recommended transformations are listed at the top of the menu with the most confident recommendation first.

The following image shows the Add Transformation menu with recommended transformations at the top:



Select a transformation from the menu to add it to the mapping in the current location.

Source recommendations

CLAIRE might recommend additional source objects when the Source transformation in a mapping uses an Amazon Redshift, Oracle, or Snowflake connection.

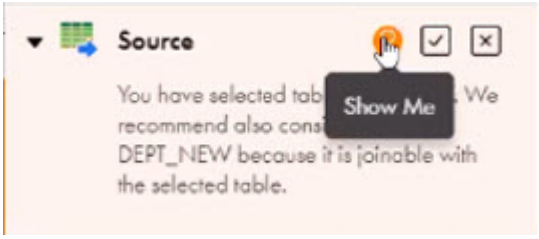
To receive CLAIRE recommendations for additional source objects belonging to a connection, the connection needs to be scanned for metadata and data. For more information about metadata extraction and data profiling using Metadata Command Center, see the Metadata Command Center help.

CLAIRE makes recommendations for additional source objects based on primary key and foreign key relationships. The recommendations can be helpful when you want to use multiple source objects and you have many tables on the data source to search through.

For example, you want to find a list of customers together with the type of car each customer has ordered. For the Source transformation in your mapping, you use a connection to an Oracle database that contains hundreds of tables. You select a customer table for the source object. In the CLAIRE **Recommendations** tab, CLAIRE suggests several tables that can be joined to the customer table. One of the tables contains customer order data. You add the table to the mapping as an additional Source transformation.

When a recommendation is available, Data Integration highlights the **Recommendations** tab. Select the **Recommendations** tab to see the recommendations.

In the list of recommendations, click the **Show Me** icon for the source that you want to investigate. A Source transformation with the recommended source object appears on the mapping canvas. The following image shows the **Show Me** icon for a recommended source:



Open the Source transformation and click the **Fields** tab to review the source fields in the source object.

If you want to use the source, in the **Recommendations** tab, click the **Accept** icon. In the mapping canvas, connect the Source transformation to the data flow.

If you don't want to use the recommended source, click **Decline**. Data Integration removes the recommended Source transformation from the mapping canvas.

Note: Source recommendations aren't shown for ORC file sources, since Data Governance and Catalog doesn't support this format.

Join recommendations

When CLAIRE recommends an additional source object, it might also recommend joining the new source and the original source with a Joiner transformation if it detects a join relationship between the two objects.

Data Integration automatically joins the sources with a normal join based on the recommended join condition. By default, when Data Integration joins the sources, it links the recommended source to the Master group and the original source to the Detail group. To avoid field name conflicts, Data Integration prefixes field names in the recommended source.

To review the recommended join condition, in the **Recommendations** tab, select **Show with Joiner** for the source that you want to review, and then click the **Show Me** icon. In the mapping canvas, open the Joiner transformation and click the **Join Condition** tab.

If you want to use the source with the Joiner transformation, in the **Recommendations** tab, click the **Accept** icon. In the mapping canvas, connect the Joiner transformation to the data flow.

Union recommendations

When CLAIRE recommends an additional source object, it might also recommend a Union transformation if it detects a union relationship between the two sources. If you accept the recommendation, Data Integration automatically unions the sources.

By default, Data Integration adds the original source as Input Group 1 and maps the original source fields to the Union transformation output fields.

To review the recommended source and Union transformation, in the **Recommendations** tab, select **Show with Union** for the source that you want to review, and then click the **Show Me** icon. If you want to use the source with the Union transformation, in the **Recommendations** tab, click the **Accept** icon. In the mapping canvas, connect the Union transformation to the data flow.

Mapping inventory recommendations

If your organization uses Enterprise Data Catalog and you use a catalog object from the inventory as a source, target, or lookup object in a mapping, CLAIRE might make recommendations for the object.

For example, CLAIRE might recommend that you add a Data Masking transformation to the mapping to mask sensitive data.

For more information about using catalog objects in mappings, see Chapter 5, “Data catalog discovery” on page 102.

Component recommendations

If your organization uses Metadata Command Center and you use a source object with columns associated with a data classification rule, CLAIRE recommends mapplets and user-defined functions associated with the same data classifications.

Before CLAIRE can recommend components, create the data classifications, configure them with a catalog source, and execute a data classification scan job in Metadata Command Center. In Data Integration, associate mapplets and user-defined functions with the data classifications. Then, when you select the source object in the Mapping Designer, CLAIRE recommends mapplets and user-defined functions based on data classifications associated with the object columns.

Mapplet recommendations

CLAIRE might recommend mapplets when the mapplets are associated with the same data classifications as the source object columns.

When a recommendation is available, Data Integration highlights the **Recommendations** tab. Select the **Recommendations** tab to see the recommendations.

In the list of recommendations, click the **Show Me** icon for the mapplet that you want to investigate. A Mapplet transformation with the recommended mapplet appears on the mapping canvas linked to the associated source.

If you want to use the mapplet, in the **Recommendations** tab, click the **Accept** icon. In the mapping canvas, connect the Mapplet transformation to the data flow.

If you don't want to use the recommended mapplet, click **Decline**. Data Integration removes the recommended Mapplet transformation from the mapping canvas.

User-defined function recommendations

CLAIRE might recommend user-defined functions when the functions are associated with the same data classifications as the source object columns.

When a recommendation is available, Data Integration highlights the **Recommendations** tab. Select the **Recommendations** tab to see the recommendations.

In the list of recommendations, select one or more user-defined functions that you want to investigate and then click the **Show Me** icon. An Expression transformation with the recommended user-defined function as a field expression appears on the mapping canvas linked to the associated source. If you select multiple user-defined functions, CLAIRE adds each user-defined function to the Expression transformation as separate field expression.

If you want to use the user-defined function, in the **Recommendations** tab, click the **Accept** icon. In the mapping canvas, connect the Expression transformation to the data flow.

If you don't want to use the recommended user-defined function, click **Decline**. Data Integration removes the recommended Expression transformation from the mapping canvas.

Data standardization recommendations

If your organization uses Metadata Command Center and you use a catalog source object, CLAIRE might recommend an Expression transformation to help you standardize your source data.

Before CLAIRE can recommend Expression transformations, you must run a profiling job on the catalog source in Metadata Command Center. Then, when you select the catalog source as your source object in the mapping, CLAIRE recommends Expression transformations based on the attributes of the source fields.

CLAIRE might recommend Expression transformations to replace NULL values or to standardize date type formats.

Replace NULL values

If CLAIRE detects a large number of fields with NULL values in the source, it might recommend an Expression transformation that replaces the NULL values with a value that you define.

CLAIRE recommends that you replace NULL values when NULL appears in more than 85% of the column fields. When a recommendation is available, Data Integration highlights the **Recommendations** tab. Select the **Recommendations** tab to see the recommendations.

In the list of recommendations, click the **Show Me** icon for the expression that you want to investigate and then enter a preview value. An Expression transformation appears on the mapping canvas linked to the associated source.

If you want to use the expression, in the **Recommendations** tab, click the **Accept** icon. Then, enter the value you want to use in place of NULL. In the mapping canvas, connect the Expression transformation to the data flow.

If you don't enter a value, CLAIRE replaces NULL values with a default value based on the data type. The following table shows the default values CLAIRE uses:

Data type	Default value
String	`default`
Integer Bgint	`0`
Date	Current system time
Decimal Double	`0.0`
Binary	`0`

If you don't want to use the recommended expression, click **Decline**. Data Integration removes the recommended Expression transformation from the mapping canvas.

Date type standardization

If CLAIRE detects multiple date formats in the source, it might recommend an Expression transformation that converts all date type fields to a common format.

The date format that CLAIRE recommends is the format occurring most frequently in the column fields. CLAIRE recommends a date format when it occurs in more than 85% of the column fields.

When a recommendation is available, Data Integration highlights the **Recommendations** tab. Select the **Recommendations** tab to see the recommendations.

In the list of recommendations, click the **Show Me** icon for the expression that you want to investigate. An Expression transformation appears on the mapping canvas linked to the associated source.

If you want to use the expression, in the **Recommendations** tab, click the **Accept** icon. In the mapping canvas, connect the Expression transformation to the data flow.

If you don't want to use the recommended expression, click **Decline**. Data Integration removes the recommended Expression transformation from the mapping canvas.

Data masking recommendations

If your organization uses Metadata Command Center and you select a catalog source object with columns that are classified as sensitive, CLAIRE recommends a Data Masking transformation to hide the sensitive data.

Before CLAIRE can recommend Data Masking transformations, create the data classifications and define the level of sensitivity in Metadata Command Center. Data classifications can have low, medium, or high sensitivity. Medium or high sensitivity mean that the column contains sensitive information.

Then, configure the classifications with a catalog source and run a data classification job to associate the data elements in the source with the data classifications.

When you select the catalog source as your source object in the mapping, CLAIRE recommends a Data Masking transformation if one or more columns in the source object have medium or high sensitivity.

Viewing data masking recommendations

When you use a source object that contains sensitive data, CLAIRE recommends a data masking transformation. When a recommendation is available, Data Integration highlights the **Recommendations** tab. You can preview the transformation and either accept or reject the recommendation.

1. To view the recommendation, open the **Recommendations** panel.
2. In the list of recommendations, expand the Data Masking transformation that you want to investigate.

   Data Integration lists the source fields with medium and high sensitivity.
3. To preview the recommendation, select the fields that you want to mask and then click the **Show Me** icon.

   A Data Masking transformation appears on the mapping canvas linked to the associated source with a default masking rule based on the column name. Data Integration also highlights the potentially sensitive fields in the source.
   You can change the default masking rule after you accept the recommendation.
4. If you want to use the Data Masking transformation, select the fields that you want to mask and then click the **Accept** icon.

   After you accept the recommendation, you can edit the default masking rule.
5. On the mapping canvas, connect the transformation to the data flow.
6. If you don't want to use the transformation, click **Decline**.

   Data Integration removes the recommended Data Masking transformation from the mapping canvas.

Rule specification recommendations

If your organization uses Data Governance and Catalog, and you use a source object with columns associated with a rule specification, CLAIRE recommends the rule specification.

Before CLAIRE can recommend rule specifications, create a data quality rule occurrence and associate a source object column to the occurrence in Data Governance and Catalog. Then, associate a rule specification to the rule occurrence and run a profiling job on the source. When you use the source object in the Mapping Designer, CLAIRE recommends the rule specification associated with the object columns.

When a recommendation is available, Data Integration highlights the **Recommendations** tab. Select the **Recommendations** tab to see the recommendations.

In the list of recommendations, click the **Show Me** icon for the rule specification that you want to investigate. A Rule Specification transformation with the recommended rule specification appears on the mapping canvas linked to the associated source. Data Integration automatically maps fields based on the field name and type.

If you want to use the rule specification, in the **Recommendations** tab, click the **Accept** icon. In the mapping canvas, connect the Rule Specification transformation to the data flow.

If you don't want to use the recommended rule specification, click **Decline**. Data Integration removes the recommended Rule Specification transformation from the mapping canvas.

CHAPTER 5

Data catalog discovery

If your organization uses Enterprise Data Catalog and you have the appropriate license, you can perform a search against the data catalog and discover catalog assets. You can use the assets that you discover as sources, targets, and lookup objects in mappings and as sources in synchronization and file ingestion and replication tasks.

Use data catalog discovery to find objects in the catalog that you can use in the following types of Data Integration assets:

- Mappings. Discover tables, views, and delimited flat files to use as sources, targets, or lookup objects in new mappings or in mappings that you currently have open in Data Integration.
- Synchronization tasks. Discover tables, views, and delimited flat files to use as sources in new synchronization tasks.
- File ingestion and replication tasks. You can discover Amazon S3, Microsoft Azure Blob Storage, and Hadoop Files objects to use as sources in file ingestion and replication tasks.

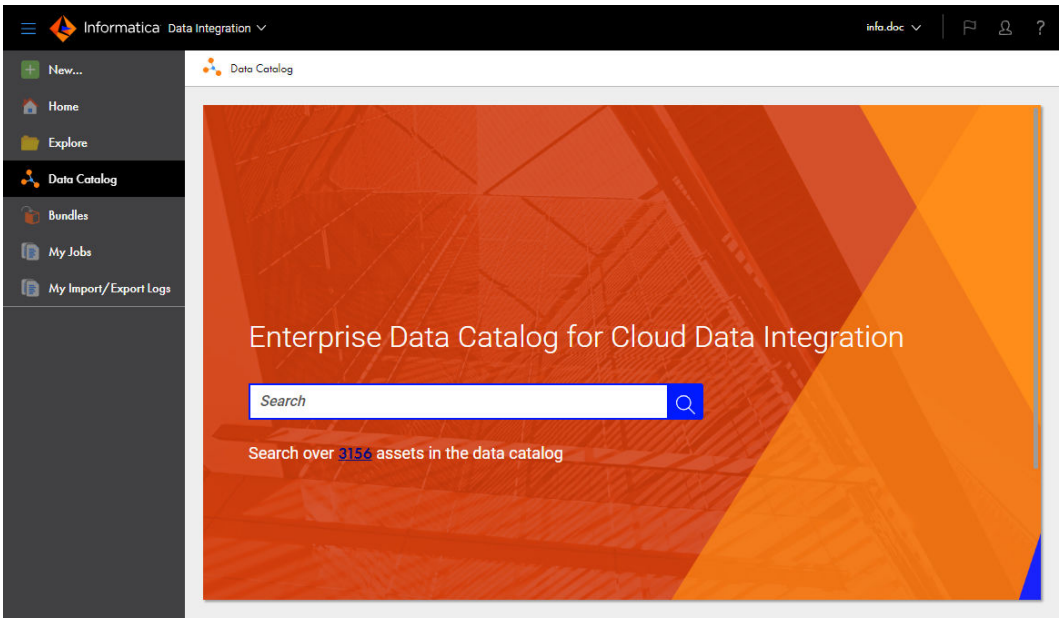
Note: Before you can use data catalog discovery, your organization administrator must configure the Enterprise Data Catalog integration properties on the **Organization** page in Administrator. For more information about configuring Enterprise Data Catalog integration properties, see the Administrator help.

You can't use data catalog discovery to find source, target, or lookup objects for mappings in SQL ELT mode.

Performing data catalog discovery

Perform data catalog discovery on the **Data Catalog** page.

The following image shows the **Data Catalog** page:



The page displays a **Search** field and the total number of table, view, and flat file assets in the catalog.

In the **Search** field, enter a phrase that might occur in the object name, description, or other metadata such as the data domain or associated business glossary term. When you select an object from the search results, Data Integration asks you where you want to use the object. Data Integration also imports the connection if it does not exist in your organization.

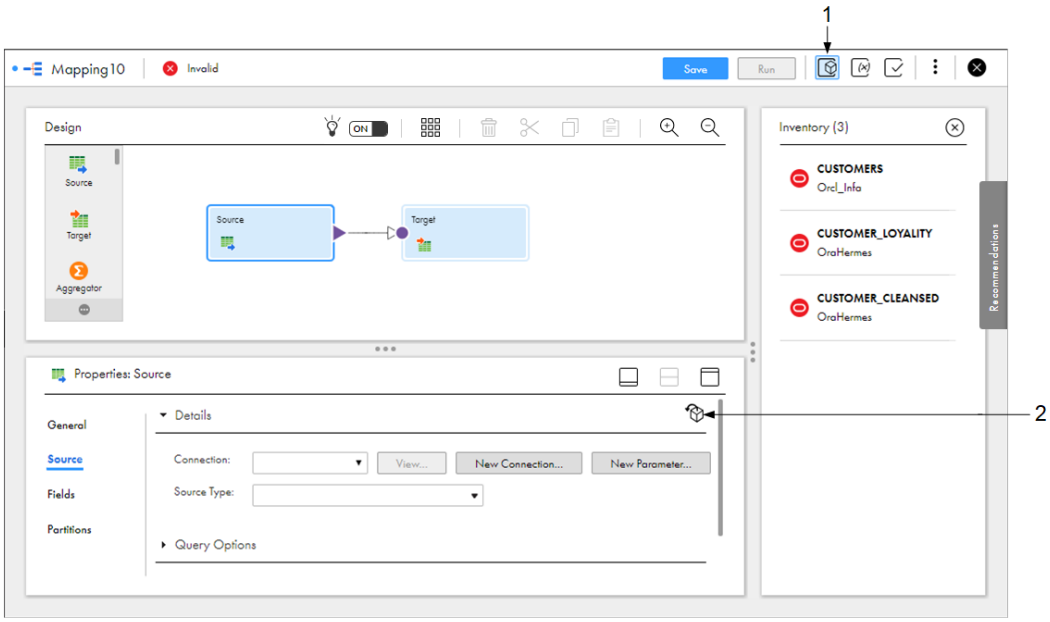
You can use the object in the following places:

- In a new mapping. If you select this option, Data Integration creates a new mapping and adds the object to the mapping inventory. You can then add the object to the mapping as a source, target, or lookup object.
- In an open mapping. If you select this option, Data Integration asks you to select the mapping and adds the object to the mapping inventory. You can then add the object to the mapping as a source, target, or lookup object.
- In a new synchronization task. If you select this option, Data Integration creates a new synchronization task and adds the object as the source object.
- In a new file ingestion and replication task. If you select this option, Data Integration creates a new file ingestion and replication task and adds the object as the source object.

Mapping inventory

If you select an object from the search results and add it to a mapping, Data Integration adds the object to the mapping inventory. Select an object from the inventory, and add it to the mapping as a source, target, or lookup object.

The following image shows the mapping inventory:



1. Inventory icon. Shows and hides the Inventory panel.
2. "Select an object from the inventory" icon in the transformation Properties panel. Selects an object from the inventory as the source, target, or lookup object.

Each mapping has its own inventory. Each object that you add to the mapping inventory stays in the inventory until you delete it.

To add an object from the inventory as a source, target, or lookup object, select the Source, Target, or Lookup transformation and click **Select an object from the inventory** on the tab where you configure the connection. Then select the object from the inventory.

If you enable CLAIRE recommendations in the Mapping Designer, Data Integration highlights the **Recommendations** tab when there is a new recommendation for objects in the inventory. For example, if an object in the inventory contains sensitive data such as credit card numbers, and you add it to the mapping as a source, target, or lookup object, Data Integration highlights the **Recommendations** tab. When you open the recommendations, CLAIRE might recommend that you add a Data Masking transformation to the mapping to mask the sensitive data.

Catalog search

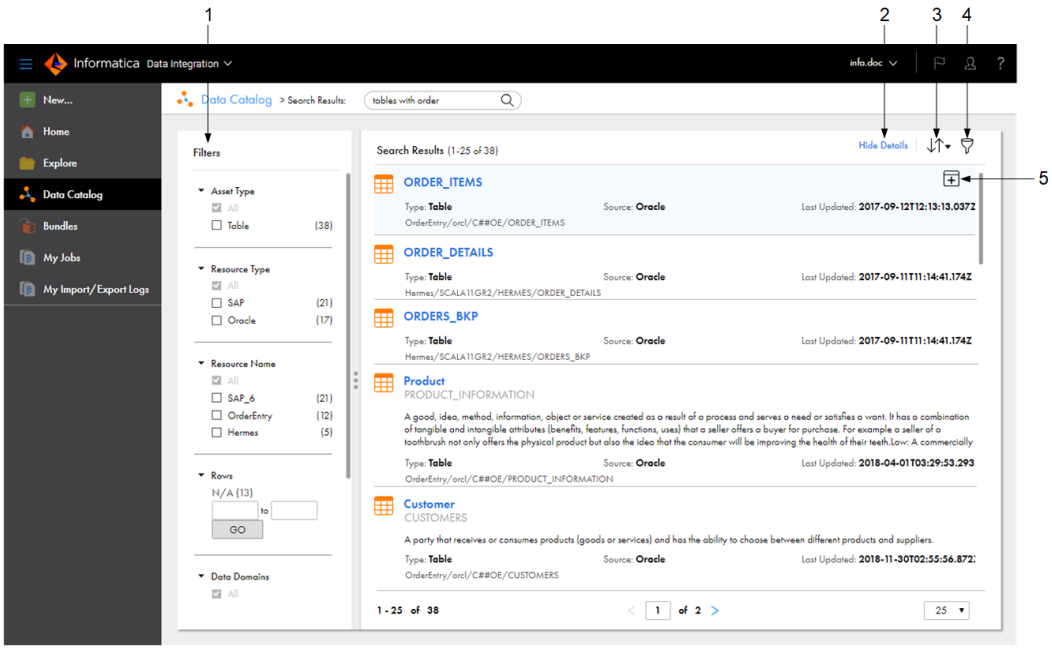
Use the search on the **Data Catalog** page to find an Enterprise Data Catalog object. Enter the object name, part of the name, or keywords associated with the object in the **Search** field, and then click the search icon. Data Integration returns all tables, views, and flat files in the catalog that match the search criteria.

You can use the `*` and `?` wildcard characters in the search phrase. For example, to find objects that start with the string "Cust", enter `Cust*` in the **Search** field.

You can also enter keyword searches. For example, if you enter `tables with order` in the **Search** field, Data Integration returns tables with "order" in the name or description, tables that have the associated business term "order," and tables that contain columns for which the "order" data domain is inferred or assigned.

For more information about Enterprise Data Catalog searches and search results, see the Enterprise Data Catalog documentation.

The following image shows an example of search results when you enter "tables with order" as the search phrase:



1. Filter search results.
2. Show or hide object details.
3. Sort search results.
4. Apply or remove all filters.
5. Use the selected object in a mapping, a synchronization task, or a file ingestion and replication task.

You can perform the following actions on the search results page:

Filter search results.

Use the filters to filter search results by asset type, resource type, resource name, number of rows, data domains, and date last updated.

Show details.

To display details about the object, click **Show Details**.

Sort results.

Use the **Sort** icon to sort results by relevance or name.

Open an object in Enterprise Data Catalog.

To open an object in Enterprise Data Catalog, click the object name. To view the object, you must log in to Enterprise Data Catalog with your Enterprise Data Catalog user name and password.

Use the object in a synchronization task, file ingestion and replication task, or mapping.

To use the object in a synchronization task, file ingestion and replication task or mapping, click **Use Object**. You can select an object if the object is a valid source, target, or lookup type for a mapping or a valid source type for the task. For example, you can select an Oracle table to use as the source in a new synchronization task, but you cannot select a Hive table.

When you select the object, Data Integration prompts you to select the task where you want to use the object and imports the connection if it does not exist.

Connection properties vary based on the object type. Data Integration imports most connection properties from the resource configuration in Enterprise Data Catalog, but you must enter other required properties, such as the connection name and password.

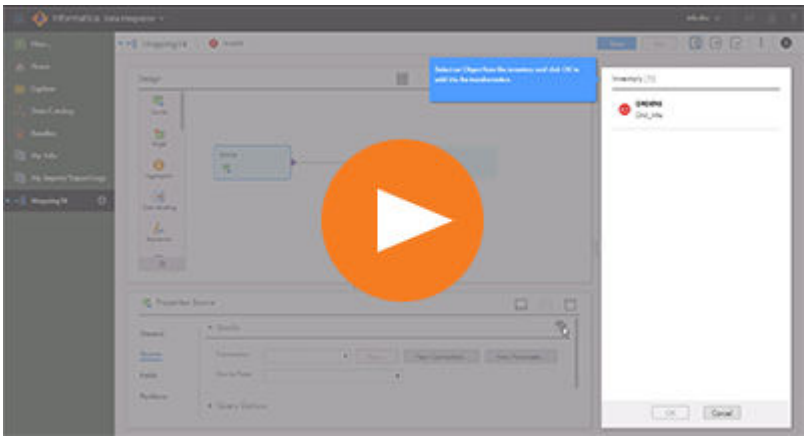
After you configure the connection or if the connection already exists, Data Integration adds the object to a new synchronization task, file ingestion and replication task, or to the inventory of a new or open mapping.

Discovering and selecting a catalog object

Discover and select a catalog object so that you can use the object as a source in a new synchronization or file ingestion and replication task, or as a source, target, or lookup object in a mapping.

Before you can use data catalog discovery, your organization administrator must configure the Enterprise Data Catalog integration properties on the **Organization** page in Administrator.

The following video shows you how to discover and select a catalog object as the source in a new mapping:



1. Open the **Data Catalog** page.
2. Enter the search phrase in the search field.

   For example, to find customer tables, you might enter "Customer," "Cust*," or "tables with customer."

3. On the search results page, click **Use Object** in the row that contains the object.

   You can select one object at a time.

   Data Integration prompts you to select where to use the object.
4. Select one of the following options:
   - To add the object to a new synchronization task, click **New Synchronization Task**.
   - To add the object to a new file ingestion and replication task, click **New File Ingestion and Replication Task**.
   - To add the object to a new mapping, click **New Mapping**.
   - To add the object to an open mapping, click **Add to an open asset**, and then select the mapping.
5. Click **OK**.

   If the connection does not exist in your organization, Data Integration prompts you to import the connection. Enter the missing connection properties such as the connection name and password.

   If you use the object in a synchronization or file ingestion and replication task, Data Integration creates the task with the object as the source. Configure other task properties such as the target, data filters, field mapping, and scheduling information.
6. If you use the object in a mapping, Data Integration adds the object to the mapping inventory. Select the object as the source, target or lookup object.

   To select the object:

   a. Open the **Inventory** panel.

   b. Select the Source, Target, or Lookup transformation.

   c. In the Properties panel, select the tab where you configure the connection:

      - In the Source transformation, click the **Source** tab.

      - In the Target transformation, click the **Target** tab.

      - In the Lookup transformation, click the **Lookup Object** tab.

   d. Click **Select an object from the inventory**.

   e. Select the object from the inventory and click **OK**.

   Data Integration adds the object to the mapping as the source, target, or lookup object.

Data catalog discovery example

You need to create a mapping that reads data from an orders table. You know that the table is in an Oracle database, but you cannot remember the exact table name or which connection the object is associated with.

You open the **Data Catalog** page and enter "tables with order" in the search field. The search returns a list of tables and views that contain “order” in the name or description, tables that have the associated business term "order," and tables that contain columns for which the "order" data domain is inferred or assigned. The search results also show details about each object.

You select the table that you were looking for and add it to a new mapping. You then open the Source transformation in the mapping, click the **Source** tab, and click **Select an object from the inventory**. You select the table from the mapping inventory and click **OK**. The table becomes the source object.

CHAPTER 6

Visio templates

A Visio template defines parameterized data flow logic used in mapping tasks.

As of the July 2024 release, Informatica has dropped support for Visio templates. You can't create new Visio templates, and you can't create or edit mapping tasks based on Visio templates. However, you can still edit Visio template properties and run existing mapping tasks based on Visio templates.

As an alternative to Visio templates, Informatica recommends that you take advantage of the mapping and mapping task templates. These templates include pre-built logic that you can use for data integration, cleansing, and warehousing tasks. The templates can be used as is or as a head start in creating a mapping or mapping task. You can choose the template that you want to use for a new mapping or mapping task on the **New Asset** dialog.

Visio template revisions

You can revise a Visio template that is used by a mapping task.

If you edit a Visio template that is used by a mapping task, Data Integration performs the following actions based on the changes that you make:

- When you change the template XML file, Data Integration lists the mapping tasks that use the Visio template and offers to keep or delete the tasks.

  Note: If you choose to delete the tasks, Data Integration deletes all listed tasks immediately. You cannot undo this action.
- When you change other elements of the Visio template, such as the input control or default value for a template parameter, Data Integration applies the changes to tasks that use the template.
  Review the tasks that use the Visio template before making changes to the template. If the changes are not compatible with an existing task configuration, the task will fail at run time.

Caution: Use caution when editing Visio templates. You can't edit the tasks that use the Visio template.

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

Configuring parameter properties

When you edit a Visio template, you can define template parameter properties, such as template parameter descriptions and display properties.

Display properties determine where and how a template parameter displays in the mapping task wizard.

If you select a different template XML file, you can choose between the following options:

- Reuse the existing mapplet and template parameter customizations.
  Data Integration uses the existing customization for mapplets and template parameters with names that match those from the previous file.
- Discard the existing mapplet and template parameter customizations.
  Data Integration clears the existing customizations.

With both options, mapplets and template parameters that are not used in the new file are deleted.

Caution: You can't edit tasks that use the Visio template. If changes to the Visio template are not compatible with the existing task configuration, the task fails.

1. To edit a Visio template, on the **Explore** page, navigate to the Visio template. In the row that contains the Visio template, click **Actions** and select **Edit**.
2. Complete the following template details:

Template details property	Description
Template Name	Name of the template.
Location	Project folder in which the Visio template resides. If the **Explore** page is currently active and a project or folder is selected, the default location for the asset is the selected project or folder. Otherwise, the default location is the location of the most recently saved asset.
Description	Optional. Description of the Visio template.
Template XML File	Visio template XML file to upload. Perform the following steps to upload a Visio template XML file: 1. Click **Select**. 2. Browse to locate and select the file you want to use, and then click **OK**.
Template Image File	Image file associated with the Visio template. Optionally, upload an image file. Use a JPG or PNG file that is less than 1 MB in size and 1024 x 768 pixels or smaller. Perform the following steps to upload an image file: 1. Click **Select**. 2. Browse to locate and select the file you want to use, and then click **OK**. To remove the selected template image file, click **Clear**.

3. If the Visio template includes a mapplet template parameter, click **Select** to select a mapplet.
4. Optionally, edit the template parameter label that appears in the mapping task wizard and provide a description of the template parameter. The description will appear as a tooltip.

5. To edit the display properties for a template parameter, click the **Edit** icon and configure the following template parameter display properties:

Template parameter display property	Required/ optional	Description
Default Value	Optional	Default value for the template parameter.
Visible	Required	Determines if the template parameter displays in the mapping task wizard. Use to hide a template parameter that does not need to be displayed.
Editable	Required	Determines if the template parameter is editable in the mapping task wizard.
Required	Required	Determines if a template parameter must be defined in the mapping task wizard.
Valid Connection Types	Required for connection template parameters	Defines the connection type allowed for a connection template parameter. Select a connection type or select **All Connection Types**.
Logical Connection	Optional	Logical connection name. Use when you want the task developer to use the same connection for logical connections with the same name. Enter any string value. Use the same string for logical connections that should use the same connection. Connection template parameters only.
Input Control	Required for string template parameters	Defines how the task developer can enter information to configure template parameters in the mapping task wizard. String template parameters only.

Field Filtering	Optional for condition, expression, and field input controls	A regular expression to limit the fields from the input control.
		Use a colon with the include and exclude statements.
		You can use a combination of include and exclude statements. Include statements take precedence.
		Use semicolons or line breaks to separate field names.
		Use any valid regular expression syntax.
		For example:
		`Include: *ID$; First_Name`
		`Last_Name`
		`Annual_Revenue`
		`Exclude: DriverID$`

Left Title	Required for field mapping input controls	Name for the left table of the field mapping display. The left table can display source, mapplet, and lookup fields.

Template parameter display property	Required/ optional	Description

Left Field Filtering	Optional for field mapping input controls	Regular expression to limit the fields that display in the left table of the field mapping display.
		Use a colon with the include and exclude statements.
		You can use a combination of include and exclude statements. Include statements take precedence.
		Use semicolons or line breaks to separate field names.
		Use any valid regular expression syntax.
		For example:
		`Include: *ID$; First_Name`
		`Last_Name`
		`Annual_Revenue`
		`Exclude: DriverID$`

Right Title	Required for field mapping input controls	Name for the right table of the field mapping display. The right table can display target, mapplet, and lookup fields.
Right Data Provider	Required for field mapping input controls	Set of fields to display in the right table of the field mapping display: - All objects. Shows all fields from all possible right table objects. - <object name>. Individual object names. You can select a single object for the right table fields to display. - Static. A specified set of fields. Allows you to define the fields to display.

Fields Declaration	Required for field mapping input controls	List of fields to display on the right table of the field mapping display.
		List field names and associated datatypes separated by a line break or semicolon (;) as follows:
		`<datatype>(<precision>,<scale>)<field name1>; <field name2>;...`
		or
		`<datatype>(<precision>,<scale>)<field name1>`
		`<field name2>`
		`<datatype>(<precision>,<scale>)<field name3>`
		If you omit the datatype, Data Integration assumes a datatype of String(255).
		Available when Static is selected for the Right Data Provider.

Right Field Filtering	Optional for field mapping input controls	Regular expression to limit the fields that display in the right table of the field mapping display.
		Use a colon with the include and exclude statements.
		You can use a combination of include and exclude statements. Include statements take precedence.
		Use semicolons or line breaks to separate field names.
		Use any valid regular expression syntax.
		For example:
		`Include: *ID$; First_Name`
		`Last_Name`
		`Annual_Revenue`
		`Exclude: DriverID$`

Template parameter display property	Required/ optional	Description
Aggregate Functions	Required for expression and field mapping input controls	Enables the display and use of aggregate functions in the **Field Expression** dialog box in the mapping task wizard.
Label	Required for custom dropdown input controls	Label for an option to display in a custom dropdown. Click **Add** to add additional options. Click **Remove** to remove any configured options.
Value	Required for custom dropdown input controls	Value for an option to display in a custom dropdown. Click **Add** to add additional options. Click **Remove** to remove any configured options.

6. To add another step to the mapping task wizard, click **New Step**.

   You can name the step and rename the Parameters step.
7. To order template parameters, use the **Move Up** and **Move Down** icons.

   You can use the icons to move connection template parameters to the Sources or Targets steps.

   You can use the icons to move template parameters from the Parameters step to any new steps.
8. If you want to add advanced session properties, click **Add**, select the property that you want to use, and configure the session property value.
9. Click **OK**.

Downloading a template XML file

You can download a template XML file for a Visio template. You might download a template XML file to upload the template into the Cloud Integration Template Designer.

1. On the **Explore** page, navigate to the Visio template.
2. In the row that contains the Visio template, click **Actions** and select **Download Template XML**.

Deleting a Visio template

Delete a Visio template when you no longer want to use the uploaded Visio template and customizations.

You cannot delete a Visio template that is used in a mapping task. Before you delete the Visio template, delete the task or replace the Visio template in the task.

1. On the **Explore** page, navigate to the Visio template.
2. In the row that contains the Visio template, click **Actions** and select **Delete**.

INDEX

A

advanced mode
  runtime plan 26
advanced session properties
  in Visio templates 108
automapping
  automapping source 31
  generate automapping 31

C

CLAIRE recommendations
  mapping design 95
Cloud Application Integration community
  URL 7
Cloud Developer community
  URL 7
components
  Visio templates 108

D

data catalog discovery
  discovering and selecting objects 106
  example 107
  mapping inventory 104
  overview 102
  performing 103
  searching for objects 105
Data flow run order 22
Data Integration community
  URL 7
data preview
  monitoring preview jobs 35
  previewing mapping data 34
  steps for previewing data 35
  viewing results 36

E

Enterprise Data Catalog
  discovering and selecting objects 106
  integration with Data Integration 102
  searching for objects 105

F

flow run order
  configuring 23

I

in-out parameters
  configuration 75
  overview 70
  properties 73
  values 74
  variable functions 72
Informatica Global Customer Support
  contact information 8
Informatica Intelligent Cloud Services
  web site 7
input parameters
  in mappings 62
  using in mappings 69
integration templates
  see Visio templates 108

M

maintenance outages 8
Mapping Designer 10
mapping tasks
  effect of mapping revisions 49
  parameters,in-out 76
  using parameter files 84
mapping templates 14
mappings
  advanced mode 23–25
  automapping 30
  configuration 17
  configuring rules and guidelines 20
  data preview 34
  data preview results 36
  in-out parameter values 74
  in-out parameters 70
  input parameters 62
  inventory panel 104
  maintenance 49
  Mapping Designer overview 10
  mapping revisions and mapping tasks 49
  overview 9
  parameters 62
  parameters,aggregation type 72
  parameters,in-out 75, 78, 80
  SQL ELT mode 26
  steps for previewing data 35
  testing 31
  tutorial 51
  using parameters 69
  validating 31
  variable functions for in-out parameters 72
mappings in SQL ELT mode
  creating 30
  mapping behavior 29

mappings in SQL ELT mode *(continued)*
  mapping configuration 29
  overview 26
  sources 27
  targets 28
  transformations 28
  validation 33
Masking tasks
  using parameter files 84

P

parameter file templates
  downloading 91
Parameter file templates 91
parameter files
  creating target object at run time 93
  location 87
  overview 84
parameters
  aggregation type 72
  configuration 67
  guidelines 75
  in mapping tasks 76
  in mappings 62
  in-out 70
  in-out parameter properties 73
  in-out parameter values 74
  in-out parameters 78, 80
  parameter types 65
  partial 68
  user defined 84
  variable functions 72

S

SQL ELT queries
  preview 41
SQL ELT query data preview
  overview 43
  running 44
SQL ELT query preview
  data preview 43
  results file 43
SQL ELT query preview *(continued)*
  running a job for mappings 42
  running a job for mappings in SQL ELT mode 42
  running data preview 44
status
  Informatica Intelligent Cloud Services 8
synchronization tasks
  using parameter files 84
system status 8

T

template XML
  downloading for Visio templates 112
templates
  mapping 14
testing mappings 31
transformations
  data preview 34
  data preview results 36
  steps for previewing data 35
trust site
  description 8

U

upgrade notifications 8
user parameters 84

V

validating mappings 31
Visio templates
  deleting 112
  downloading template XML 112
  overview 108
  revising 108

W

web site 7