



Informatica® Data Integration - Free & PayGo
April 2023

Mappings

© Copyright Informatica LLC 2022, 2023

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: 2023-04-04

Table of Contents

Preface	5
Informatica Resources.	5
Informatica Documentation.	5
Informatica Intelligent Cloud Services web site.	5
Informatica Intelligent Cloud Services Communities.	5
Informatica Intelligent Cloud Services Marketplace.	5
Data Integration connector documentation.	6
Informatica Knowledge Base.	6
Informatica Intelligent Cloud Services Trust Center.	6
Informatica Global Customer Support.	6
 Chapter 1: Mappings.....	7
Mapping Designer.	8
Mapping templates.	9
Mapping configuration.	11
Defining a mapping.	12
Configuring the source.	12
Configuring the data flow.	13
Configuring the target.	14
Rules and guidelines for mapping configuration.	14
Data flow run order.	15
Mapping validation.	16
Validating a mapping.	17
Data preview in mappings.	17
Preview behavior for a mapping.	18
Running a preview job for a mapping.	18
Viewing preview results for a mapping.	19
Customizing preview results.	20
Detailed data preview in mapplets.	21
Running detailed data preview in a mapplet.	21
Pushdown optimization preview.	22
Running a pushdown preview job.	23
Pushdown optimization preview results files.	23
Pushdown optimization data preview.	23
Running a pushdown data preview job.	24
Field lineage.	24
Lineage for renamed fields.	25
Lineage for mapped fields.	26
Mapplet field lineage.	26
Lineage for lookup fields.	27

Lineage for transformations that read Data Quality assets.	27
Viewing field lineage.	28
Testing a mapping.	28
Mapping maintenance.	29
Mapping revisions and mapping tasks.	29
Chapter 2: Parameters.	30
Input parameters.	30
Input parameter types.	32
Input parameter configuration.	34
Partial parameterization with input parameters.	35
Using parameters in a mapping.	37
In-out parameters.	37
Aggregation types.	38
Variable functions.	39
In-out parameter properties.	40
In-out parameter values.	40
Rules and guidelines for in-out parameters.	41
Creating an in-out parameter.	42
Editing in-out parameters in a mapping task.	42
In-out parameter example.	44
Using in-out parameters as expression variables.	46
Parameter files.	47
Parameter file requirements.	47
Parameter scope.	48
Sample parameter file.	49
Parameter file location.	50
Rules and guidelines for parameter files.	52
Parameter file templates.	52
Overriding connections with parameter files.	53
Overriding data objects with parameter files.	54
Overriding source queries.	54
Creating target objects at run time with parameter files.	55
Chapter 3: CLAIRE recommendations.	56
Transformation type recommendations.	57
Source recommendations.	58
Join recommendations.	58
Union recommendations.	59
Index.	60

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 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, go to <https://status.informatica.com/> and click **SUBSCRIBE TO UPDATES**. You can then 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 Customer Support Center by telephone or online.

For online support, click **Submit Support Request** in Informatica Intelligent Cloud Services. You can also use Online Support to log a case. Online Support requires a login. You can request a login at <https://network.informatica.com/welcome>.

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 data loader or data transfer tasks, such as specific ordering of logic.

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.

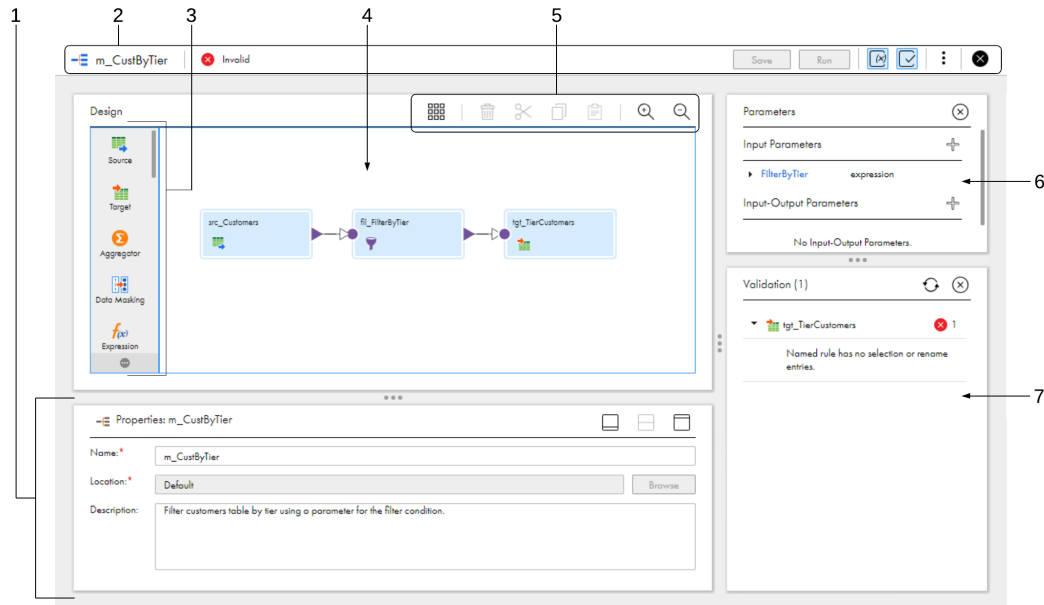
You can configure parameters to enable additional flexibility in how you can use the mapping. Parameters act as placeholders for information that you define in the mapping task. For example, you can use a parameter for a source connection in a mapping, and then define the source connection when you configure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

You can use components such as mapplets, shared sequences, and user-defined functions in mappings. Components are assets that support mappings. For example, a mapplet is reusable transformation logic that you can use in mappings. A shared sequence is a reusable sequence that you can use in multiple Sequence Generator transformations.

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:

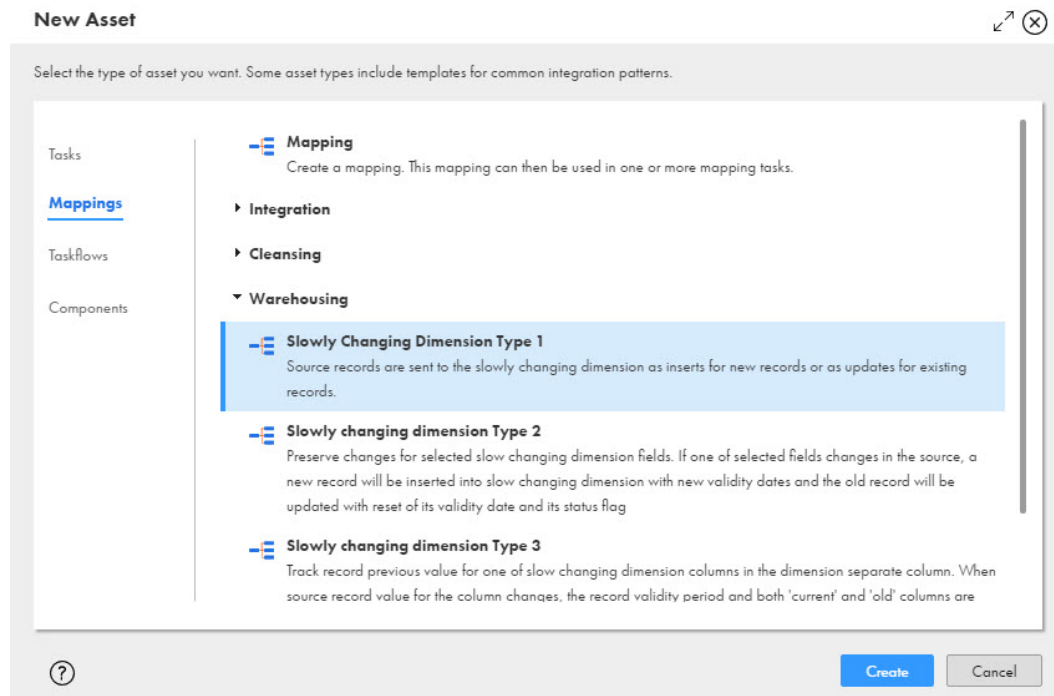
Mapping Designer Areas	Description
1. Properties panel	<p>Displays configuration options for the mapping or selected transformation. Different options display based on the transformation type.</p> <p>Includes icons to quickly resize the Properties panel. Use the icons to display the Properties panel, the mapping canvas, or both.</p> <p>You can also manually resize the Properties panel.</p>
2. Header	<p>Displays the following information, buttons, and icons:</p> <ul style="list-style-type: none"> - Mapping name. - Mapping status. The status can be either Valid or Invalid. - 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. - Parameters icon. Displays and hides the Parameters panel. - Validation icon. Displays and hides the Validation panel. - Pushdown Optimization icon. Displays and hides the Pushdown Optimization panel. - Actions icon. Use to create and save a mapping task based on the mapping. See <i>Tasks</i> for more information about mapping tasks. - Close icon.
3. Transformation palette	<p>Lists the transformations that you can use in the mapping.</p> <p>To add a transformation, drag the transformation to the mapping canvas.</p>
4. Mapping canvas	<p>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.</p>

Mapping Designer Areas	Description
5. Toolbar	<p>Displays the following icons and buttons:</p> <ul style="list-style-type: none"> - 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. Parameters panel	<p>Lists the parameters in the mapping. You can create, edit, and delete parameters, and see where the mapping uses parameters.</p> <p>Displays when you click Parameters. To hide the panel, click Parameters again.</p>
7. Validation panel	<p>Lists the transformations in the mapping and displays details about mapping errors. Use to find and correct mapping errors.</p> <p>Displays when you click Validation. To hide the panel, click Validation again.</p>

Mapping templates

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

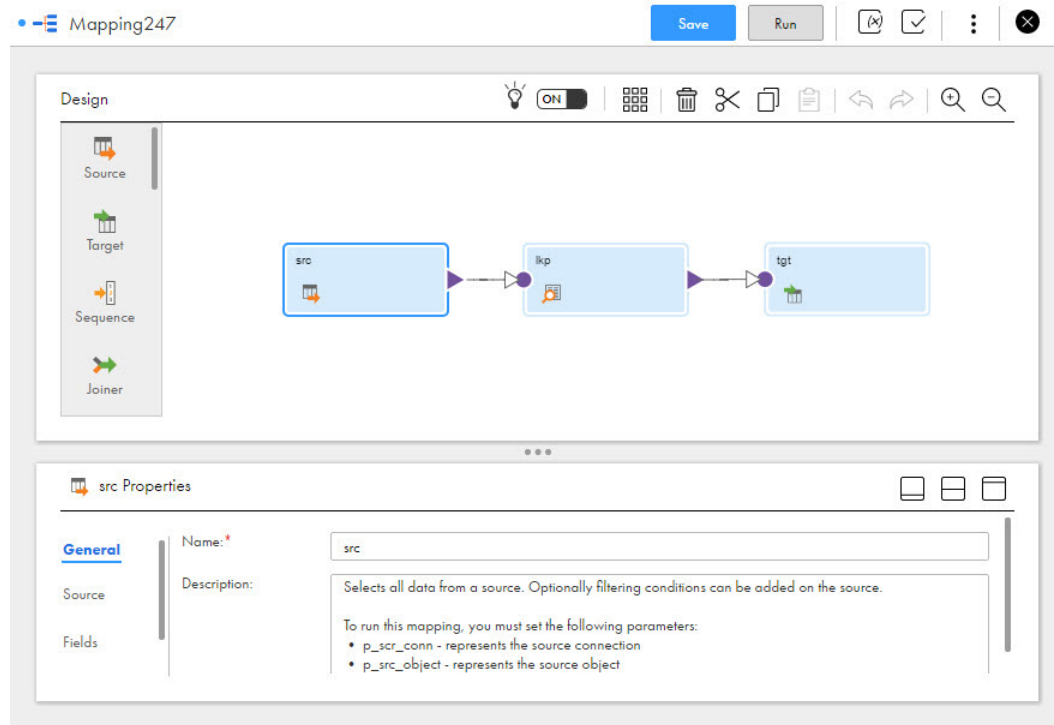
Mapping templates are divided into the categories: Integration, Cleansing, and Warehousing.



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

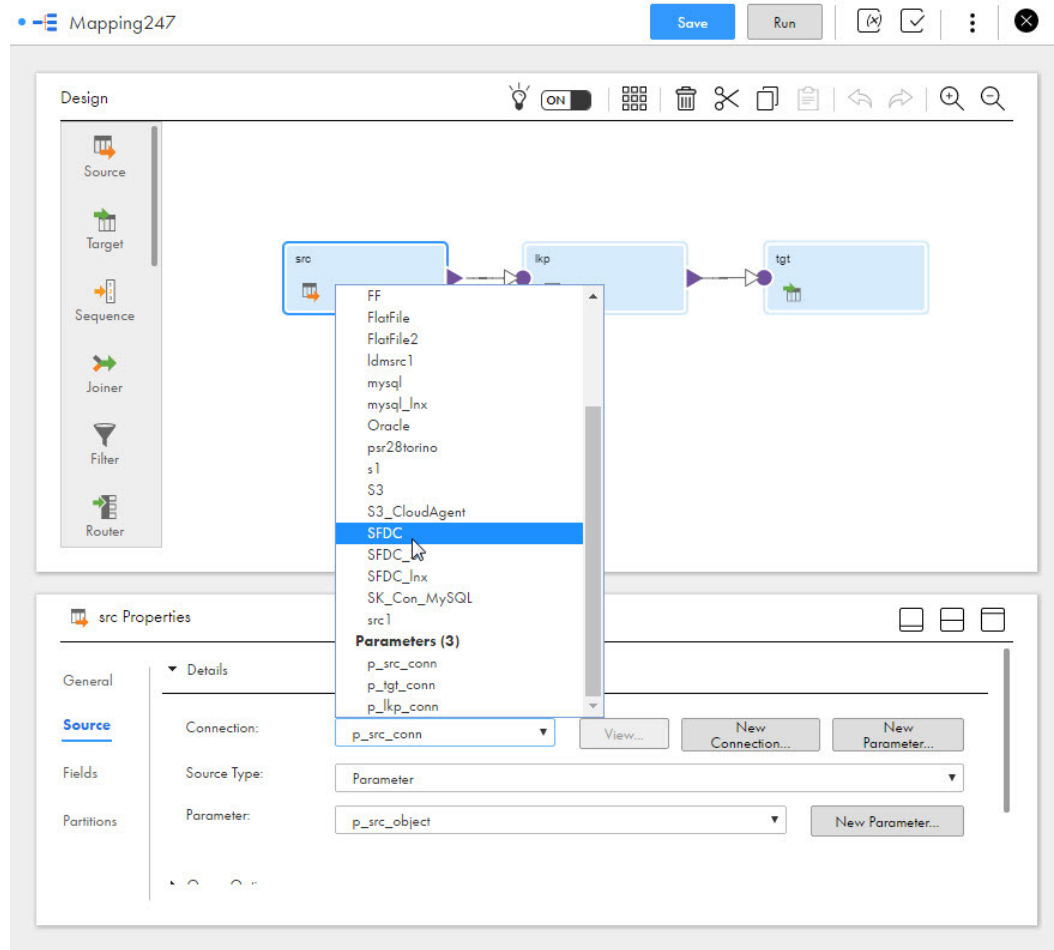
The mapping contains pre-populated transformations. Click on each of the transformations in the mapping to see the purpose of the transformation, how the transformation is configured, and which parameters are used.

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_src_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.

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

1. Click **New > Mappings**, and then perform one of the following tasks:
 - To create a mapping from scratch, click **Mapping** and then click **Create**. The Mapping Designer appears with a Source transformation and a Target transformation in the mapping canvas for you to configure.
 - To create a mapping based on a template, click the template you want to use and then click **Create**. The Mapping Designer appears with a complete mapping that you can use as is or you can modify.
 - 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 appears with the mapping that you selected.
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 *Transformations*.

In the source details, select the source connection and source object. For some connection types, you can select multiple source objects. You can also configure parameters for the source connection and source object.

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.
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 *Transformations*.
In the target details, select the target connection, target object, and target operation. 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.
- 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.

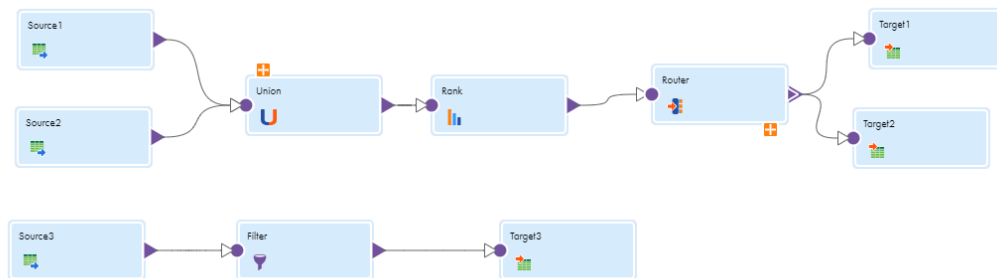
Data flow run order

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.

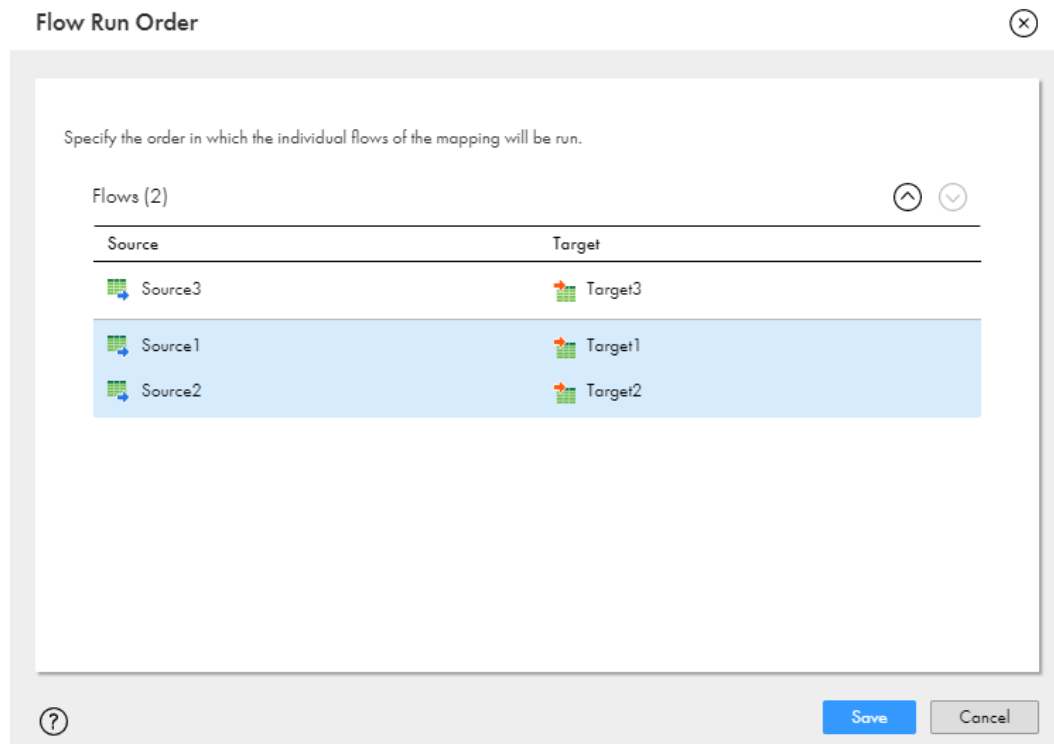
If a flow contains multiple targets, you cannot configure the load order of the targets within the flow.

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 pushdown optimization in a mapping that uses an ODBC connection.

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*.

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**.

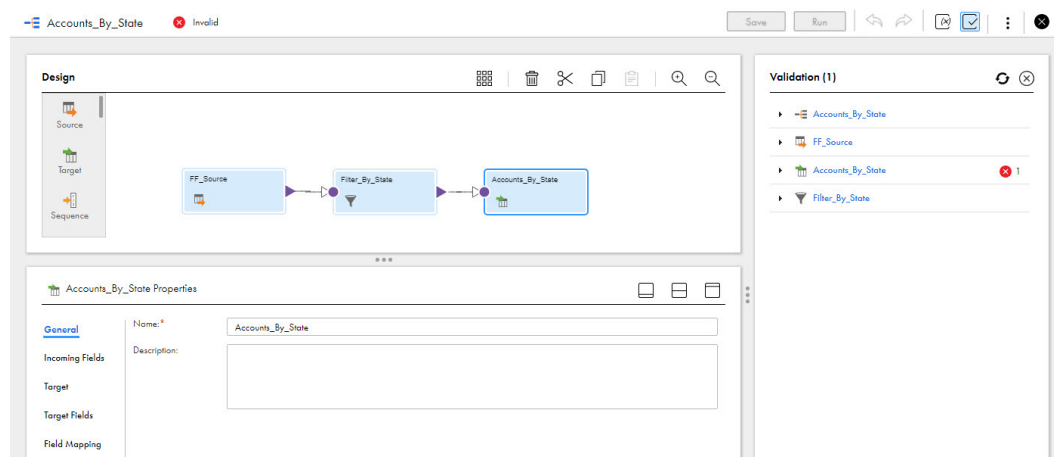
Mapping validation

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

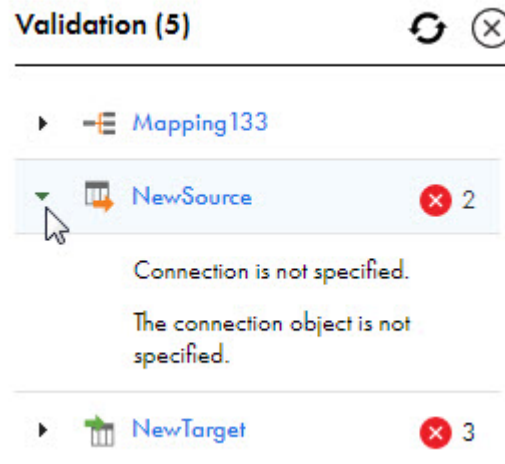
Use the **Validation** panel to view error details.

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.

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:

The screenshot shows the Data Integration Maplet Designer interface. The top bar includes a title 'm_FilterAndSortCustRecords', a 'Valid' status, and buttons for 'Save' and 'Run'. The 'Design' panel shows a sequence of transformations: 'src_AllCust', 'fil_ByTier', 'srt_ByCustID', and 'tgt_Cust'. The 'Preview' panel is active, displaying a table of data for the 'srt_ByCustID' transformation. The table has columns for Customer, Customer, Lastname, Firstname, Company, Address1, Address2, Address3, City, City2, State, Zip, PrefDeliv, Country, CreateDa, Currency, and OrderAn. The data is as follows:

Customer	Customer	Lastname	Firstname	Company	Address1	Address2	Address3	City	City2	State	Zip	PrefDeliv	Country	CreateDa	Currency	OrderAn
19134129	Diamond	Keating	Sandra	FIRST ...	5742 H...			GARDE...		CA	92845	US Post...	US	22/10...	USD	36336
19134119	Diamond	Buck	Chris	JOSEP...	200 PA...			NEW Y...		NY	10166	Federal...	US	14/10...		26798
191341...	Diamond	Bennett	David	STATUE...	52 VA...			NEW Y...		NY	10017...	United ...	US	31/08...	USD	32831
191341...	Diamond	Walker	Sharon	FEDER...	10 EXC...			JERSEY ...		NJ	07302	Federal...	US	26/08...	USD	47450
191340...	Diamond	Drodge	Herbert	MILLEN...	666 FIF...			NEW Y...		NY	10005	United ...	US	9/7/1...	USD	50159

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:

- Sequence Generator
- 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 cannot 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.

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.

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 *Secure Agent Services*.

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.

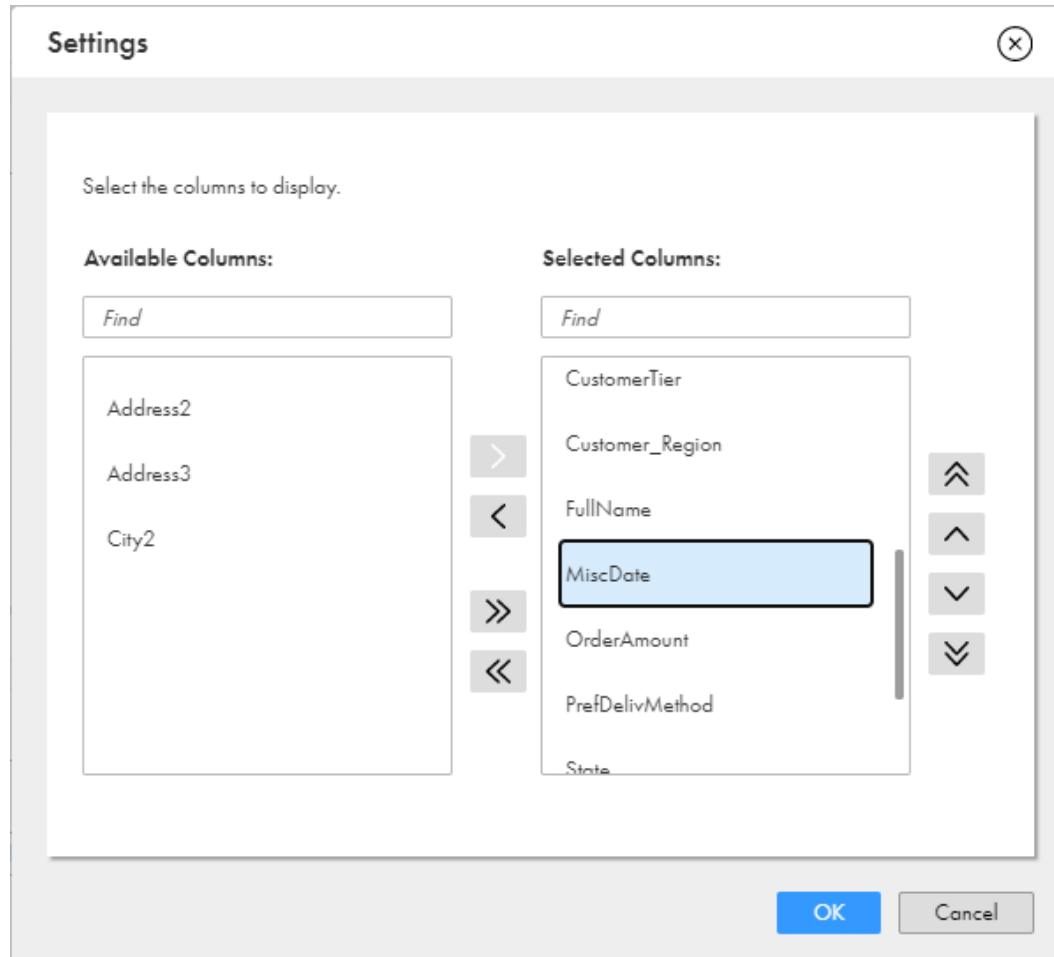
Data Integration purges the preview directory once every 24 hours. During the purge, Data Integration deletes files that are more than 24 hours old.

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.

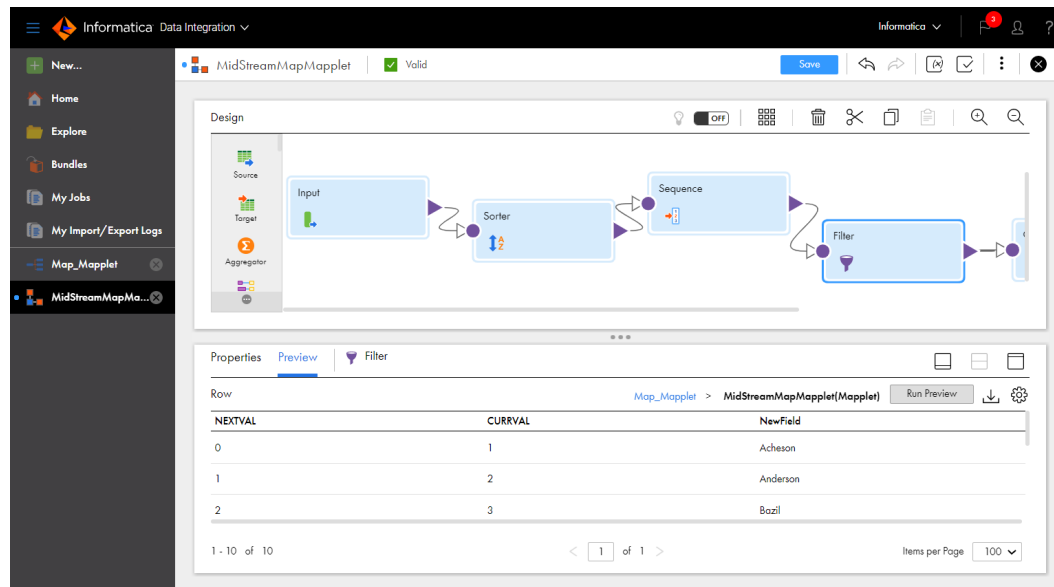
Detailed data preview in mapplets

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:

- Input
- Output
- Sequence Generator
- Target

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 mapplet is valid.

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

1. In the mapping, open the preview panel for the Maplet transformation.
2. Click **Detailed Preview in Maplet**.
The maplet opens.
3. In the maplet, 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 maplet, click the asset name in the breadcrumb.

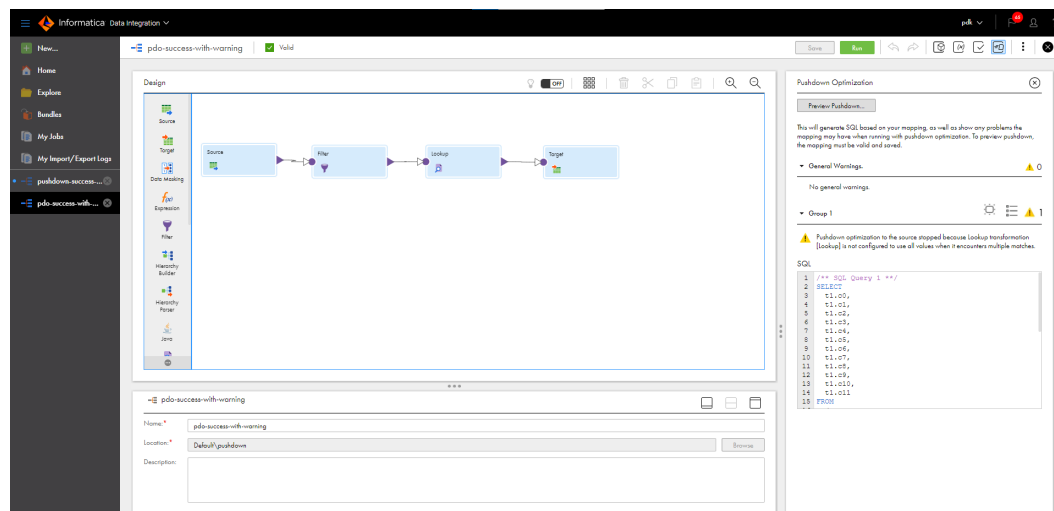
Pushdown optimization preview

When you create a mapping that is configured for pushdown optimization, you can preview the SQL queries that Data Integration pushes to the database. You can also preview the data after the queries are executed. Preview pushdown optimization in the **Pushdown Optimization** panel in the Mapping Designer.

You can preview pushdown optimization results for some connector types. For more information, see the help for the appropriate connector.

When you preview pushdown optimization, Data Integration creates and runs a temporary pushdown preview mapping task. When the job completes, Data Integration displays the SQL to be executed and any warnings in the **Pushdown Optimization** panel. Data Integration groups SQL and warnings based on the data flow run order. After you run the pushdown preview job, you can preview the data after it is transformed by the transformations included in the SQL queries.

The following image shows the **Pushdown Optimization** panel:



If the pushdown optimization type that you select is not available, Data Integration lists the SQL queries, if any, that can be executed. For example, if you select Full pushdown optimization, but the target does not support pushdown, Data Integration displays the SQL queries that will be pushed to the source.

Running a pushdown preview job

Preview the SQL queries that Data Integration pushes to the database on the **Pushdown Optimization** panel.

Before you run pushdown optimization preview, verify that the following conditions are true:

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

To run a pushdown optimization job:

1. Open the **Pushdown Optimization** panel.
2. Click **Preview Pushdown**.
3. In the Pushdown 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 pushdown optimization options.
6. Click **Pushdown Preview**.

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

To view the transformations included in pushdown 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**, or **All Jobs** pages. Data Integration names the job <mapping name>_pdo_preview-<instance number>, for example, Mapping1_pdo_preview-2. You can download the session log for the preview job.

If you update the mapping after you run a pushdown preview job, the preview is no longer valid. To restart a preview job, run it again from the **Pushdown Optimization** panel. You cannot restart the job from the **My Jobs**, **Running Jobs**, or **All Jobs** pages.

Pushdown optimization preview results files

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

If you run the 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 *Secure Agent Services*.

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

Pushdown optimization data preview

After you run a pushdown optimization preview job, you can preview the data after it is transformed by the transformation logic included in the SQL queries. You can preview the first 15 rows of data for any group in the **Pushdown Preview** panel.

When you run a pushdown optimization data preview job, Data Integration creates a temporary mapping task that contains a virtual target immediately before the target or the last transformation possible for pushdown optimization.

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

The following table describes the transformation that data preview runs on for each type of pushdown.

Pushdown type	Transformation
Full	Transformation immediately before the Target transformation.
Partial	Last transformation where pushdown is possible.

You can't preview pushdown optimization data for the following transformations:

- Router
- Sequence Generator
- Target

You can't preview pushdown data for a mapping if the last transformation where pushdown is possible is a Mapplet transformation.

If you run a pushdown data preview job at the same time you run a mapping data preview job in the same mapping, the job fails.

Running a pushdown data preview job

To preview the data after the pushdown SQL queries are executed, run data preview. You run pushdown data preview after you run pushdown optimization preview.

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 **Pushdown Optimization Data Preview** window opens.
3. To download the data preview results as a CSV file, click **Download**.

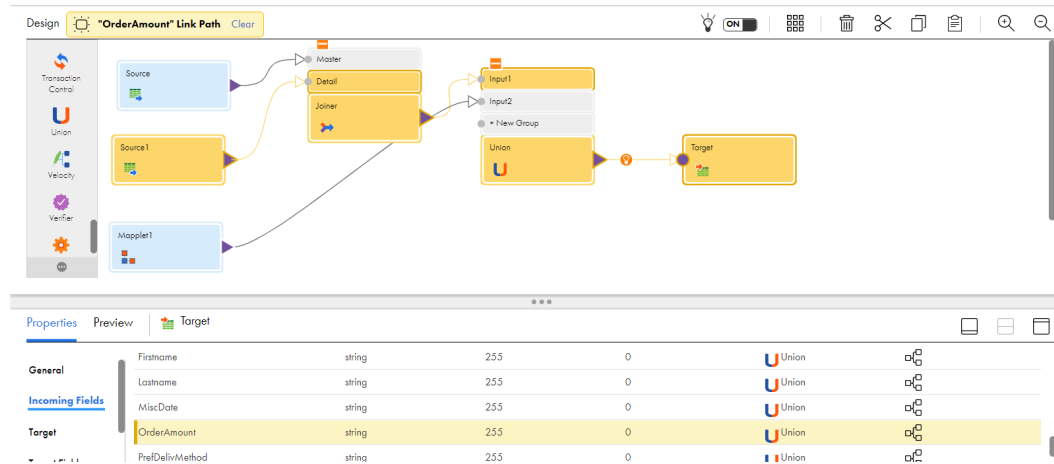
Field lineage

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:

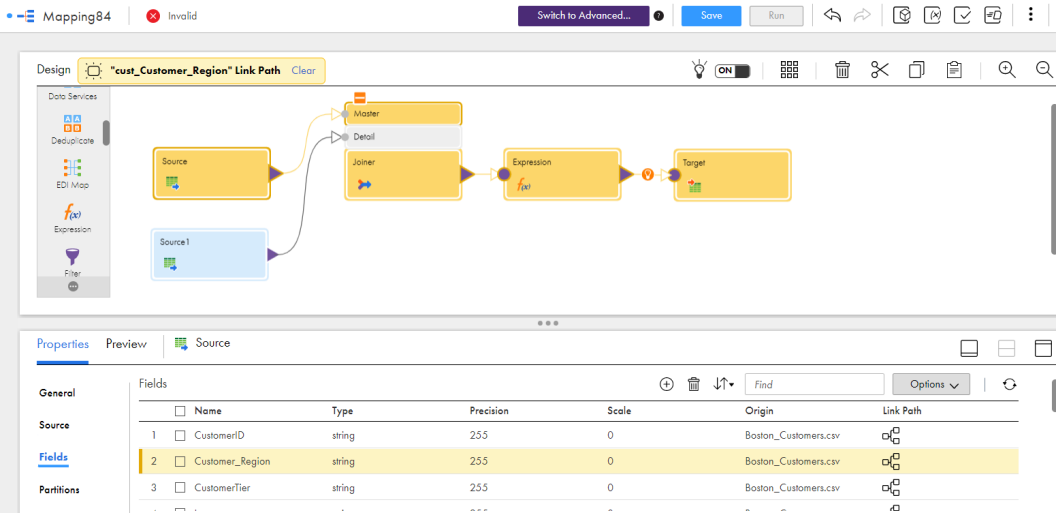
- Hierarchy Builder
- Hierarchy Parser
- Hierarchy Processor
- Java
- Mapplet transformations that reference an SAP or PowerCenter mapplet
- Python
- SQL transformations that process an SQL query
- Unconnected transformations

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 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`.

Maplet field lineage

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

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

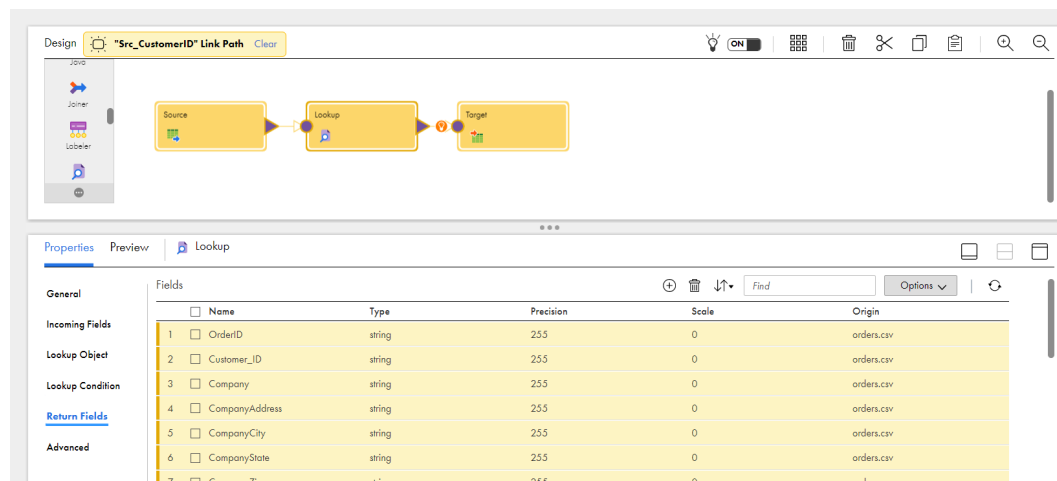
If the Mapplet transformation references an SAP or PowerCenter mapplet, field lineage stops at the Mapplet transformation.

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.

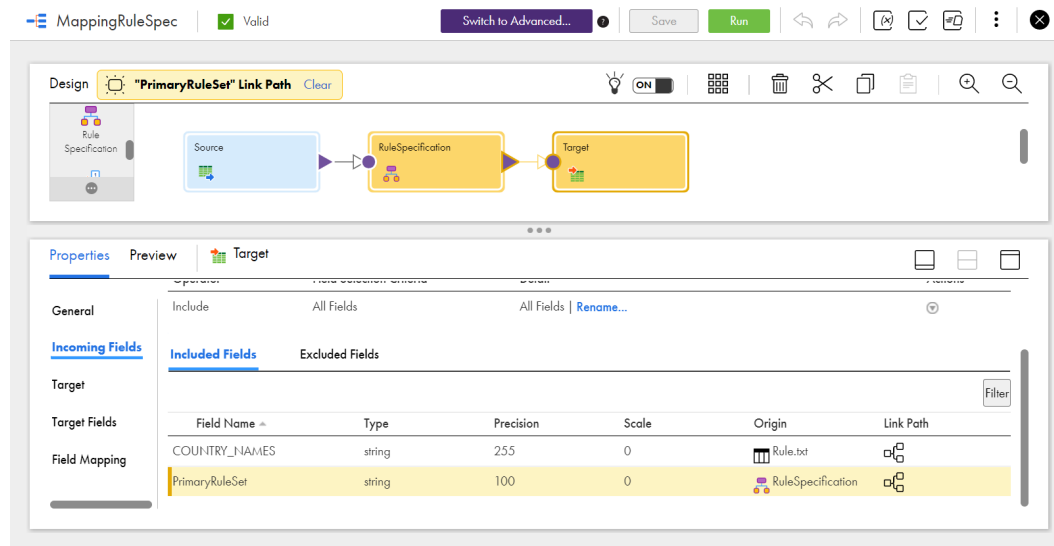
Cleanse transformations

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*.

Mapping maintenance

You can view, configure, copy, 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.

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.

CHAPTER 2

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	<p>You can use an input parameter for the following parts of the Source transformation:</p> <ul style="list-style-type: none"> - 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	<p>You can use an input parameter for the following parts of the Target transformation:</p> <ul style="list-style-type: none"> - 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: <ul style="list-style-type: none"> - 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	<p>You can use an input parameter for the following parts of the Incoming Fields tab of any transformation:</p> <ul style="list-style-type: none"> - 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	<p>You can use an input parameter for the following parts of the Aggregator transformation:</p> <ul style="list-style-type: none"> - 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.
Expression	<p>You can use an input parameter for an expression in the Expression transformation.</p> <p>In the task, you create the entire expression.</p>
Filter	<p>You can use an input parameter for the following parts of the Filter transformation:</p> <ul style="list-style-type: none"> - 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	<p>You can use an input parameter for the following parts of the Joiner transformation:</p> <ul style="list-style-type: none"> - 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.
Lookup	<p>You can use an input parameter for the following parts of the Lookup transformation:</p> <ul style="list-style-type: none"> - 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.

Transformation	Input parameter use in mappings and tasks
Mapplet	<p>You can use an input parameter for the following parts of the Mapplet transformation:</p> <ul style="list-style-type: none"> - 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: <ul style="list-style-type: none"> - 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. <p>You can configure input parameters separately for each input group.</p>
Rank	<p>You can use an input parameter for the number of rows to include in each rank group.</p> <p>In the task, you enter the number of rows.</p>
Router	<p>You can use an input parameter for the following parts of the Router transformation:</p> <ul style="list-style-type: none"> - 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	<p>You can use an input parameter for the following parts of the Sorter transformation:</p> <ul style="list-style-type: none"> - 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	<p>You can use an input parameter for the following parts of the SQL transformation:</p> <ul style="list-style-type: none"> - 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.
Union	<p>You can use an input parameter for the following parts of the Union transformation:</p> <ul style="list-style-type: none"> - 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: <ul style="list-style-type: none"> - 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. <p>You can configure input parameters separately for each input group.</p>

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

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

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

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 Maplet transformation
- Field mapping in the Target transformation

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.
Type	Parameter type. Determines where you can use the parameter. Also determines how the parameter displays in a mapping task: <ul style="list-style-type: none"> - 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:

The screenshot shows the 'Query Options' dialog box with the 'Filter' tab selected. The 'Filter Condition' dropdown is set to 'Advanced'. Below it, the 'Fields' list contains 'PARAM_FIELD' and 'PARAM_SORT'. A 'Filter Condition' text box contains 'city = \$PARAM_FIELD\$'. Below the text box is an 'Operators' row with buttons for AND, OR, NOT, (,), =, !=, <, >, <=, and >=. At the bottom, there are fields for Name, Label, Default Value, Type, and Description. The 'Name' field is set to 'PARAM_FIELD' and the 'Type' is set to 'Field'. The 'OK' and 'Cancel' buttons are at the bottom right.

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.

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.

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

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.

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.	Max or Min	All transformation data types.
SetMaxVariable	Sets the parameter to the maximum value of a group of values.	Max	All transformation data types.
SetMinVariable	Sets the parameter to the minimum value of a group of values.	Min	All transformation data types.
SetCountVariable	Increments the parameter value by one.	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.
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.
Default Value	Optional. Default value for the parameter, which might be the initial value when the mapping first runs. Use the following format for default values for datetime variables: MM/DD/YYYY HH24:MI:SS.US.
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: <ul style="list-style-type: none">- On success or warning- On success- On warning- 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: <ul style="list-style-type: none">- 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:

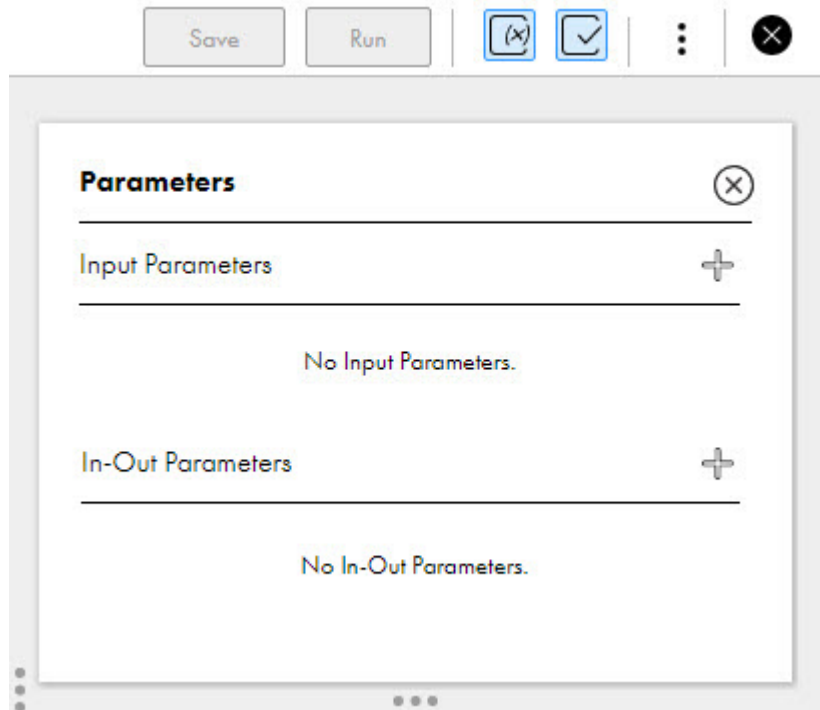
- When you write expressions that use in-out parameters, you don't need string identifiers for string variables.
- When you use a parameter in a transformation, enclose string parameters in string identifiers, such as single quotation marks, to indicate the parameter is a string.
- When you use in-out 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 required, change the format of a date/time parameter to match the format in the source.
- If you copy 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.
- 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.
- You can't use data preview 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 40](#) and [“Mapping Designer” on page 8](#).

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:

Dialog box titled "Edit mt_LoadCustomersToFF" with tabs: 1 Definition, 2 In-Out Parameters (selected), 3 Schedule.

Buttons: Reset All, Save, < Back, Next >, Finish, Cancel.

Action	Name	Desc	Type(prec, scale)	Default Value	Value	Retention Policy	Aggregation Type
	Timestamp	Date/Time the mapping was run.	datetime(29 9)	06/01/2017 12:00:00 0000000001	06/28/2017 11:24:15.0000000000	On success or warning	Max

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:

Job Properties:

- Task Name: mt_LoadCustomersToFF
- Task Type: Mapping Task
- Started By: troy
- Start Time: Jun 28, 2017 11:24:08 AM
- End Time: Jun 28, 2017 11:24:16 AM
- Duration: 8 seconds
- Runtime Environment: CAW184178
- Secure Agent: CAW184178

Results:

- Status: Success
- Success Rows: 1999
- Error Rows: 0
- Session Log: [Download Session Log](#)

Individual Source/Target Results:

Name	Success Rows	Error Rows
Customers_In	1999	0
Customers_Out	1999	0

In-Out Parameters:

Name	Value
\$\$Timestamp	06/28/2017 11:24:15.0000000000

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.

Note: You can also configure a date/time data type if your source uses a date format like YYYY-MM-DD HH:MM:SS. In that case, use the `SetMaxVariable` function.

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

The screenshot shows the 'New In-Out Parameter' dialog box. The 'Name' field is filled with 'IncludeMaxDate'. The 'Data Type' dropdown is set to 'string'. The 'Precision' field is '40' and the 'Scale' field is '0'. The 'Is expression variable' checkbox is unchecked. The 'Default Value' field contains '2017-01-01'. The 'Retention Policy' dropdown is set to 'On success or warning'. The 'Aggregation Type' dropdown is set to 'Max'. The dialog has 'OK' and 'Cancel' buttons at the bottom right.

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.

New Field ⓧ

Create new output field, variable field, input macro field or output macro field.

Field Type: Output Field ▼

Name: * OutMaxDate

Type: * string ▼

Precision: * 40

Scale: 0

? OK Cancel

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.

Field Expression: OutMaxDate(string, 40, 0) ⓧ

Configure expression by adding fields and functions.

Expression: Not Parameterized ▼

Fields Parameters Functions

- id
- first
- last
- email
- phone
- access_level
- timestamp

Expression Validate

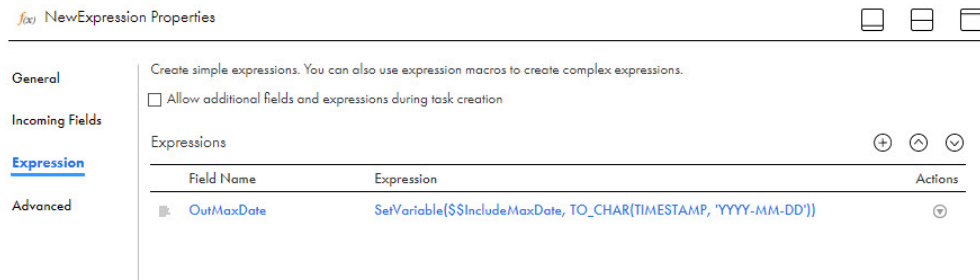
SetVariable(\$\$IncludeMaxDate, TO_CHAR(TIMESTAMP, 'YYYY-MM-DD'))

Operators

AND OR NOT () = != < > <= >=

? OK Cancel

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.

Using in-out parameters as expression variables

You can use an in-out parameter as a placeholder for an expression. To use an in-out parameter as an expression, create a string parameter and enable the **Is expression variable** option.

When you enable this option, 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.
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.

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 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.

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

Parameter file requirements

You can reuse parameter files across assets such as mapping tasks and 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, and taskflows.

Heading	Description
[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].[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 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 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 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.

By default, Data Integration uses the following parameter file directory on the Secure Agent machine:

```
<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.

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.

System variable	Description
\$PMStorageDir	Directory for files related to the state of operation of internal processes such as session and workflow recovery 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 pmrtdm.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: <ul style="list-style-type: none"> - 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: <ul style="list-style-type: none"> - 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: <ul style="list-style-type: none"> - 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.
- 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."

- 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.

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.

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.

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
```

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 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 3

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.

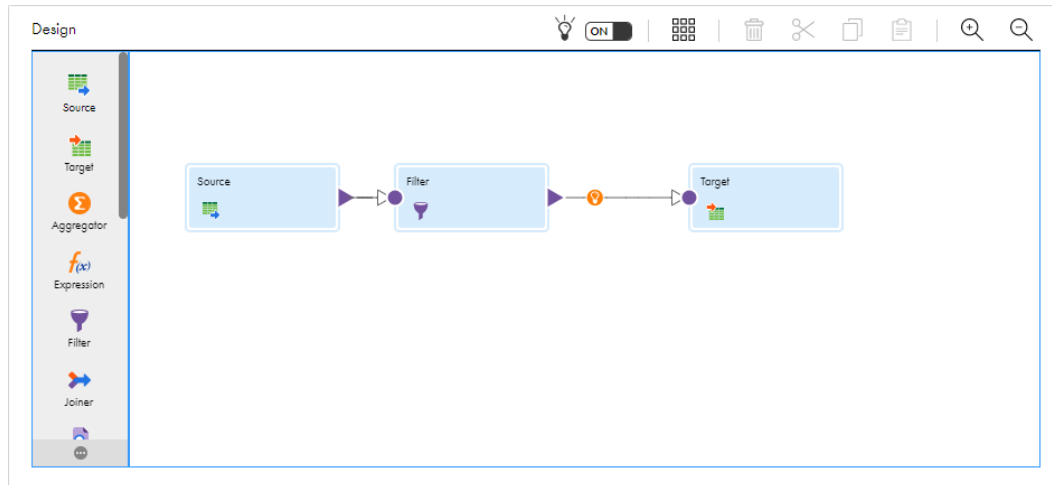
CLAIRE can make the following types of recommendations during mapping design:

- 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.

Transformation type recommendations

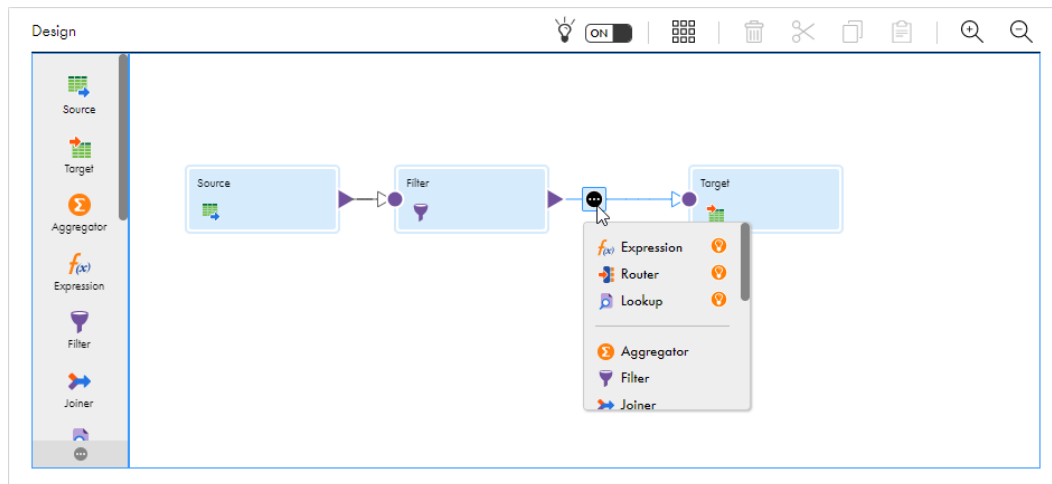
CLAIRE uses design metadata and the current flow of your mapping to recommend transformations in the data flow. CLAIRE polls the mapping after every change to provide the most relevant recommendations.

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.

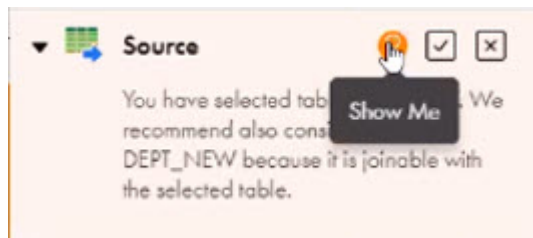
Source recommendations

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

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.

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.

INDEX

C

CLAIRE recommendations
mapping design [56](#)
Cloud Application Integration community
URL [5](#)
Cloud Developer community
URL [5](#)

D

Data flow run order [15](#)
Data Integration community
URL [5](#)
data preview
monitoring preview jobs [18](#)
previewing mapping data [17](#)
steps for previewing data [18](#)
viewing results [19](#)

F

flow run order
configuring [16](#)

I

in-out parameters
configuration [42](#)
overview [37](#)
properties [40](#)
values [40](#)
variable functions [39](#)
Informatica Global Customer Support
contact information [6](#)
Informatica Intelligent Cloud Services
web site [5](#)
input parameters
in mappings [30](#)
using in mappings [37](#)

M

maintenance outages [6](#)
Mapping Designer [8](#)
mapping tasks
effect of mapping revisions [29](#)
parameters,in-out [42](#)
using parameter files [47](#)
mapping templates [9](#)
mappings
configuration [11](#)

mappings (*continued*)
configuring rules and guidelines [14](#)
data preview [17](#)
data preview results [19](#)
in-out parameter values [40](#)
in-out parameters [37](#)
input parameters [30](#)
maintenance [29](#)
Mapping Designer overview [8](#)
mapping revisions and mapping tasks [29](#)
overview [7](#)
parameters [30](#)
parameters,aggregation type [38](#)
parameters,in-out [41](#), [44](#)
steps for previewing data [18](#)
testing [16](#)
using parameters [37](#)
validating [16](#)
variable functions for in-out parameters [39](#)
Masking tasks
using parameter files [47](#)

P

parameter file templates
downloading [53](#)
Parameter file templates [53](#)
parameter files
creating target object at run time [55](#)
overview [47](#)
parameters
aggregation type [38](#)
configuration [34](#)
guidelines [41](#)
in mapping tasks [42](#)
in mappings [30](#)
in-out [37](#)
in-out parameter properties [40](#)
in-out parameter values [40](#)
in-out parameters [44](#)
parameter types [32](#)
partial [35](#)
user defined [47](#)
variable functions [39](#)
Pushdown optimization
preview [22](#)
pushdown optimization preview
results file [23](#)
running a job [23](#)
Pushdown optimization preview [22](#)

S

status

Informatica Intelligent Cloud Services [6](#)

synchronization tasks

using parameter files [47](#)

system status [6](#)

T

templates

mapping [9](#)

testing mappings [16](#)

transformations

data preview [17](#)

data preview results [19](#)

steps for previewing data [18](#)

trust site

description [6](#)

U

upgrade notifications [6](#)

user parameters [47](#)

V

validating mappings [16](#)

W

web site [5](#)