

Mapplets in Informatica Cloud (IICS)

1. Introduction

2. What are Mapplets in Informatica Cloud?

3. Mapplet Input and Output

3.1. Mapplet Input

3.2. Mapplet Output

4. Steps to create a Mapplet

5. Mapplet Transformation

6. Demonstration: Implementing Deduplication logic using a Mapplet

Step-1: Create a Mapplet with Deduplication logic

Step-2: Configure Mapplet transformation in the mapping to insert mapplet

Step-3: Configure Mapping to pass data and read output from the mapplet

7. Conclusion

1. Introduction

In the process of building mapping components in a project, we come across scenarios where a certain business rule or logic is applied commonly across a set of mappings. Instead of building the logic separately for each mapping, Informatica supports embedding the logic into a reusable object called Mapplet and use it across several mappings with same requirement.

In this article, let us discuss in-detail about how to create Mapplets and use them in [Informatica Cloud](#) mappings.

2. What are Mapplets in Informatica Cloud?

A Mapplet is a reusable object that consists of a set of transformations which can be used across mappings to transform data before it is loaded into the target.

Creating a mapplet object is similar to creating a mapping. Once a mapplet is created, it can be used in a mapping using Mapplet transformation. Multiple mapplets can be used in a single mapping.

Note that you can use a mapplet in another mapplet. However, the mapplets cannot reference each other recursively. For example, if Mapplet A uses Mapplet B, Mapplet B cannot use Mapplet A.

3. Mapplet Input and Output

Apart from the reusable transformation logic, you must configure the Input and Output for a mapplet.

A Mapplet must contain at least one Input transformation or one Output transformation.

3.1. Mapplet Input

A Mapplet input can be an Input transformation, a Source transformation, or both.

- Use an Input transformation when you want the mapplet to receive input data from upstream transformations.
- Use Source transformation in a mapplet to provide source data.
- When only a Source transformation is used as a mapplet input, the mapplet should be the first object in the mapping pipeline and contains no input groups.

A Mapplet must contain at least one Input transformation or Source transformation.

3.2. Mapplet Output

A Mapplet output can be to an Output transformation, a Target transformation, or both.

- Use an Output transformation when you want the mapplet to pass data to a downstream transformation.
- Use a Target transformation when you want the mapplet to write data to a target.
- When you use a Target transformation with no Output transformation, the mapplet should be the last object in the mapping pipeline.

A Mapplet must contain at least one Output transformation or Target transformation.

4. Steps to create a Mapplet

Follow below steps to create a mapplet in Informatica Cloud

1. Navigate to **New > Mapplets > Mapplet**.
2. The Mapplet Designer appears with an Input transformation and an Output transformation on the canvas.
3. Specify the mapplet *Name* and *Location*.
4. Configure the mapplet input. Input can be a Source transformation, an Input transformation, or both. If you use an Input transformation, add input fields for the fields you want to pass to the mapplet.
5. Add transformations to the mapplet and configure the logic.
6. Configure the mapplet output. Output can be a Target transformation, an Output transformation, or both. If you use an Output transformation, add output fields for the data you want to pass to the downstream transformation.
7. Validate and Save the mapplet.

5. Mapplet Transformation

Mapplet Transformation inserts the mapplet object created in [Data Integration](#) into a mapping. A mapplet transformation can insert only a single mapplet. Multiple mapplet transformations can be used in a mapping to use multiple mapplets.

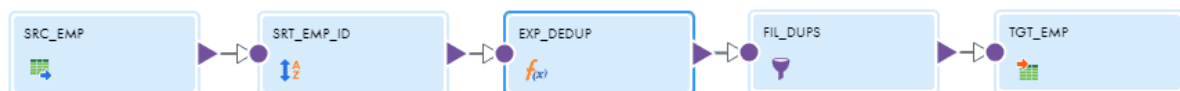
Mapplet transformation can be either Active or Passive based on the transformation logic used in the mapping.

A Maplet transformation is active if atleast one active transformation is used in the maplet and can return different number of rows compared to source rows passed to the maplet.

A Maplet transformation is passive when only passive transformation are used in the maplet and returns same number of rows as source rows passed to the maplet.

6. Demonstration: Implementing Deduplication logic using a Maplet

A common scenario we come across in [data integration](#) projects is the presence of duplicates in the source data. The duplicate records can be easily filtered out in Informatica Cloud mappings by [comparing current record with previous record](#).



Mapping with Deduplication logic implemented in Expression

The above mapping is one such example where,

- The Employees data is used as source.
- The data is then sorted based on EMPLOYEE_ID using Sorter transformation.

Sort: Not Parameterized ▼

Sort Conditions ⊕ ⌶ ⌵

Field	Sort Order
EMPLOYEE_ID	Ascending

Sorting the data on EMPLOYEE_ID before checking for duplicates

- The deduplication logic is implemented in an Expression transformation. Each record is flagged whether it a unique or a duplicate record.

Expressions

Field Name	Expression	Type	Precision	Scale
V_FLAG	IIF(EMPLOYEE_ID != V_EMPLOYEE_ID, 1, 0)	integer	10	0
V_EMPLOYEE_ID	EMPLOYEE_ID	string	100	0
O_FLAG	V_FLAG	integer	10	0

Implementing Deduplication logic in Expression

- Based on the flag created in expression, the duplicate records are filtered out using filter transformation.

Filter Condition: Advanced ▼

O_FLAG = 1

Selecting only Unique records using Filter

- The clean data is finally loaded into the target.

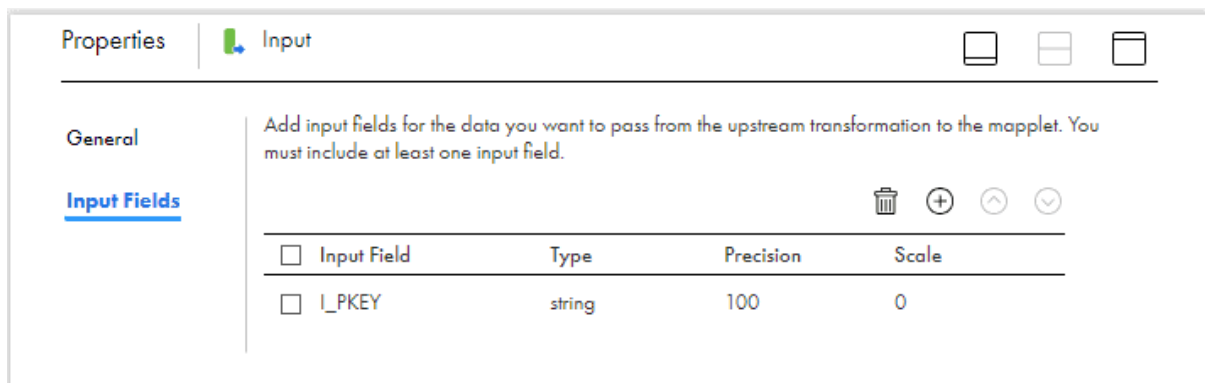
In this demonstration, we will replace the expression transformation holding the deduplication logic with a maplet. The advantage of using a maplet to implement deduplication logic is, it could be used across multiple mappings and reduces the burden of implementing the same logic with different fields in each mapping.

Follow below steps to introduce a maplet transformation with deduplication logic in the mapping.

Step-1: Create a Maplet with Deduplication logic

1. To create a maplet, navigate to **New > Maplets > Maplet**.
2. Configure the Input transformation in the maplet. Go to **Input Fields** section of the transformation and click **+** to add a new input field.

The Input field reads data coming from upstream transformation in a mapping.



Properties | Input

General

Input Fields

Add input fields for the data you want to pass from the upstream transformation to the maplet. You must include at least one input field.

<input type="checkbox"/> Input Field	Type	Precision	Scale
<input type="checkbox"/> I_PKEY	string	100	0

Buttons: [Trash] [Add] [Up] [Down]

Creating Input field in Input Transformation in Maplet

3. Add an expression transformation after Input transformation and configure deduplication logic using the field created in Input transformation.

Properties

EXP_DEDUP

General

Incoming Fields

Expression

Window

Advanced

Create simple expressions. You can also use expression macros to create complex expressions.

☐ Allow additional fields and expressions during task creation

Expressions

Field Name	Expression	Type	Precision	Scale
V_FLAG	IIF(I_PKEY != V_PKEY, 1, 0)	integer	10	0
V_PKEY	I_PKEY	string	100	0
O_FLAG	V_FLAG	integer	10	0

Implementing Deduplication logic in mapplet using the Input Field

4. Pass data from expression to target transformation. Go to **Output Fields** section of the transformation and click + to add a new output field.

The Output field passes the data to the downstream transformation in a mapping.

Properties

Output

General

Incoming Fields

Output Fields

Field Mapping


Add output fields for the output data you want to pass to the downstream transformation. You must include at least one output field.

☐ Output Field

Type	Precision	Scale	
O_FLAG	integer	10	0

Creating Output Filed in Output Transformation

5. Under **Field Mapping** section, Map the output flag field created in the expression transformation to the output field created in Output transformation

Properties |  Output

General | Field map options: Manual Options

Incoming Fields | **Incoming Fields: (1 of 2 ...)** Find

Field Name ^
O_FLAG
I_PKEY

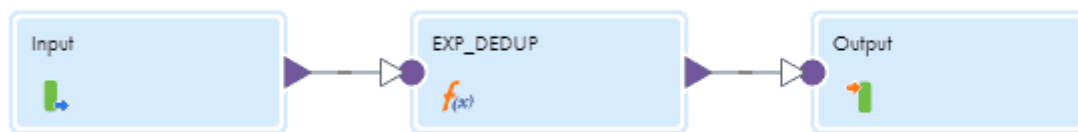
Output Fields | **Output Fields: (1 of 1 mapp...** Find Automap ⋮

Field Name ^	Mapped Field
O_FLAG	O_FLAG

Field Mapping

Field Mapping in Output Transformation

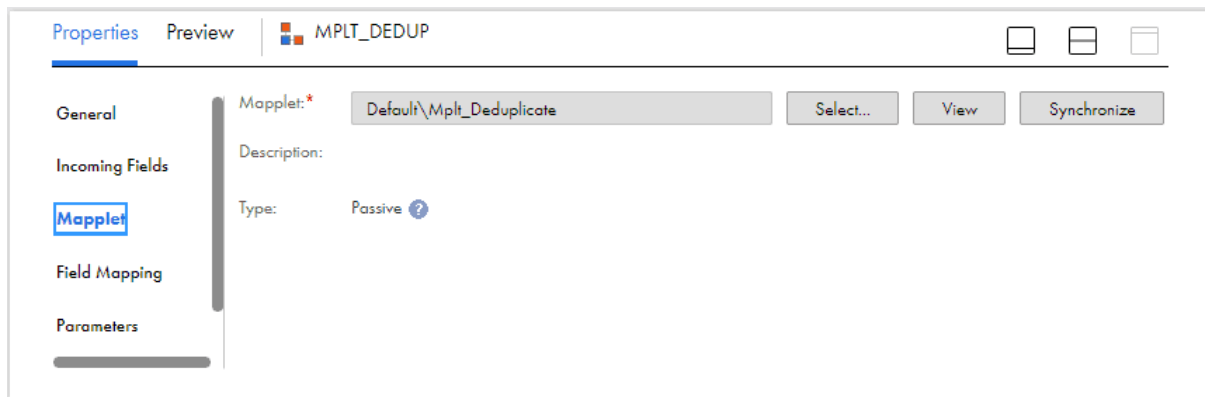
6. Validate and Save the mapplet.



Mapplet with Deduplication Logic

Step-2: Configure Mapplet transformation in the mapping to insert mapplet

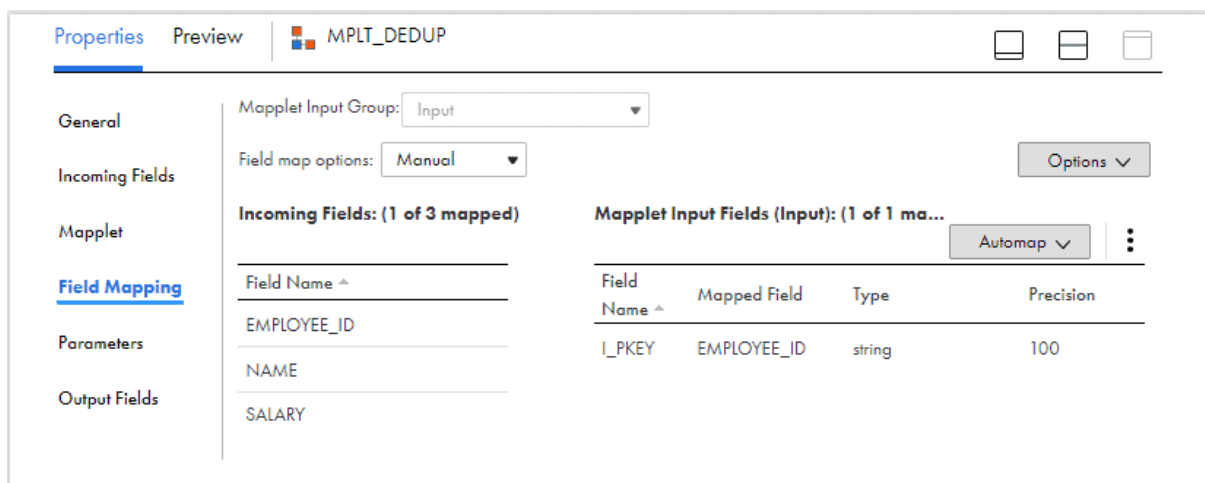
1. Go to mapping in which deduplication logic to be implemented and drag and drop the Mapplet transformation into the mapping canvas.
2. In the mapplet transformation, under the **Mapplet** section, select the mapplet created in earlier step.



Selecting Maplet in a Maplet transformation

Note that any changes done in the maplet are not propagated to all the instances of the maplets used in the mappings. The user needs to manually sync the changes using “Synchronize” button.

3. Map the data from Sorter transformation to the Maplet transformation.
4. In the Maplet transformation, under Field Mapping section, map the EMPLOYEE_ID field to the input field coming from maplet.



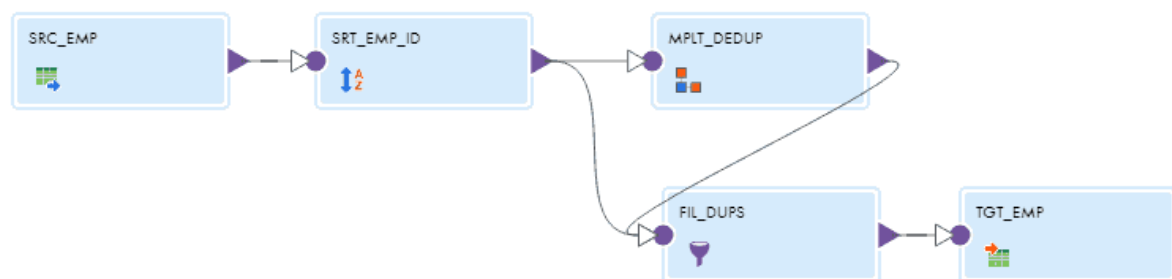
Mapping Source Fields to Maplet Input Fields in Maplet Transformation

Note that if there are multiple Inputs configured in a Mapplet, they can be selected using the “Mapplet Input Group”.

Step-3: Configure Mapping to pass data and read output from the mapplet

1. Pass the data from Mapplet transformation to the filter transformation which filters out the duplicates.
2. The mapplet is configured to give only one output which is a flag field indicating if the record is a duplicate or not.

In order to pass the source fields towards target, pass the data from sorter transformation to filter transformation using another link as shown below.



Mapping with Deduplication logic implemented using Mapplet

Note that mapping data from multiple transformations to a single transformation is “NOT” allowed by Informatica if the mapplet is active.

3. Validate and Save the mapping.

The Deduplicate Mapplet can be used in multiple mappings to remove duplicates. Simply pass the primary key field of your source as input to mapplet and use the output to identify the duplicates. Additionally, you can also implement NULL check in the same mapplet and perform both NULL and DEDUPLICATE check on different source fields and then flag the record

accordingly. This is super useful and acts as an “Ultimate Data Quality Check component” in your [data integration](#) processes.

7. Conclusion

Mapplets are a powerful feature to build reusable transformation logic and use across multiple mappings.

To conclude the discussion on mapplets,

1. Mapplets help in building complex logic once and reuse it across mappings.
2. Mapplets help in hiding the complex logic. Mapplets only show Input and Output fields. The transformation logic used is not displayed in the mapping.
3. If there are any changes done in a mapplet, the mappings and the mapplets that use the mapplet are invalid until the mapplet is synchronized.