

Informatica PowerCenter

Lesson 5: Mapplets

Lesson Objectives

- In this lesson you will learn about:
 - Concept of Mapplet
 - Mapplet Input and Output Transformations
 - Configuring a Normalizer Transformation



5.1. Concept of Mapplet

Description

- A mapplet is a reusable object that represents a set of transformations
- It allows to reuse transformation logic and can contain as many transformations as needed

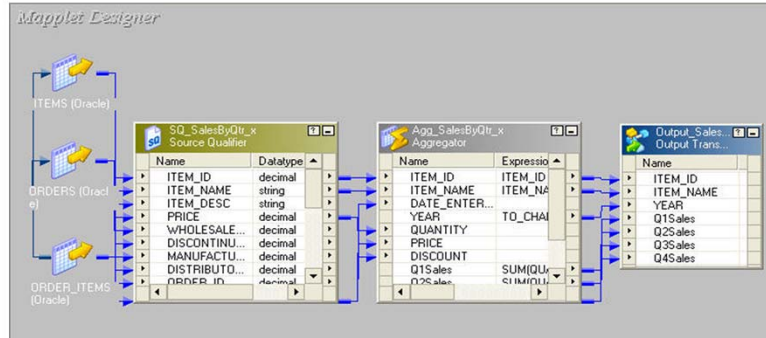


Figure 5.1

A mapplet is a reusable object that is created in the Mapplet Designer. It contains a set of transformations. It allows to reuse the transformation logic in multiple mappings.

5.1. Concept of Mapplet

What can Mapplets do?

- Mapplets help simplify mappings in the following ways:
 - Include source definitions
 - Accept data from sources in a mapping
 - Include multiple transformations
 - Pass data to multiple transformations
 - Contain unused ports



Copyright © Capgemini 2015. All Rights Reserved 4

Mapplets help simplify mappings in the following ways:

- **Include source definitions** - Multiple source definitions and source qualifiers can be used to provide source data for a mapping
- **Accept data from sources in a mapping** - The mapplet can receive source data from the mapping by using an Input transformation to receive source data
- **Include multiple transformations** - A mapplet can contain as many transformations as needed
- **Pass data to multiple transformations** - A mapplet can feed data to multiple transformations. Each Output transformation in a mapplet represents one output group in a mapplet
- **Contain unused ports** - Like a reusable transformation, a mapplet can have input and output ports that are not used in a mapping. This helps to design a mapplet for a range of uses

5.2. Mapplet Input and Output

Description

- To use a mapplet in a mapping it has to be configured for input and output
- In addition to transformation logic a mapplet has the following components:
 - Mapplet input
 - Mapplet output
 - Mapplet ports
- Targets cannot be placed in a mapplet
- It has two additional transformations
 - Mapplet Input (Optional)
 - Mapplet Output



Copyright © Capgemini 2015. All Rights Reserved 5

In addition to the transformation logic, a mapplet has the following components.

- **Mapplet input** - Data can be passed into a mapplet using source definitions and/or Input transformations. An Input transformation is used to connect a mapplet to the source pipeline in the mapping
- **Mapplet output** - Each mapplet must contain one or more output transformations to pass data from the mapplet into the mapping
- **Mapplet ports** - Mapplet ports display only in the Mapping Designer. Mapplet ports consist of input ports from Input transformations and output ports from Output transformations. Other ports within the mapplet are referred to as transformation ports. If a mapplet uses source definitions rather than Input transformations for input, it does not contain any input ports in the mapping

5.2. Mapplet Input and Output

Mapplet Input Transformation

- A Mapplet Input Transformation is used when input has to be received from a source in a mapping.(Source definitions are in Mapping)
- It provides input ports to pass data through the mapplet.

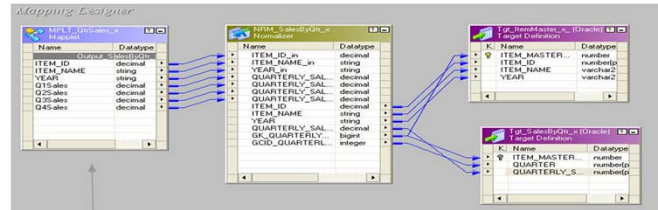


Figure 5.2

Mapplet Input Transformation

An Input Transformation is used to receive input from a source in a mapping. When the mapplet is used in a mapping, the Input transformation provides input ports so data can be passed through the mapplet. Each port in the Input transformation connected to another transformation in the mapplet becomes a mapplet input port. Input transformations can receive data from a single active source.

5.2. Maplet Input and Output

Maplet Output Transformation

- A Maplet Output Transformation is used to pass data through the maplet into a mapping
- A maplet must contain at least one Output transformation with at least one connected port in the maplet

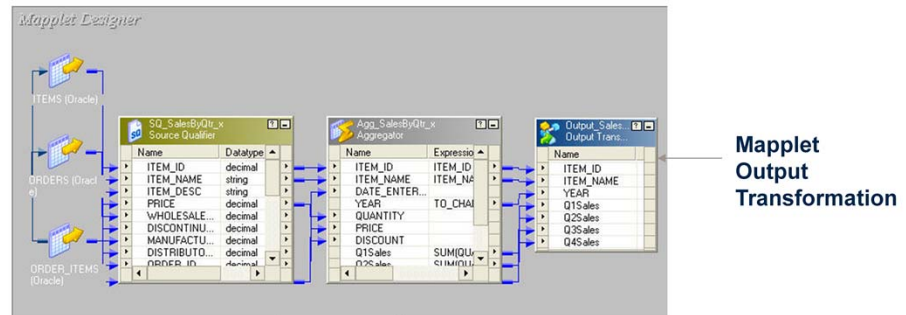


Figure 5.3

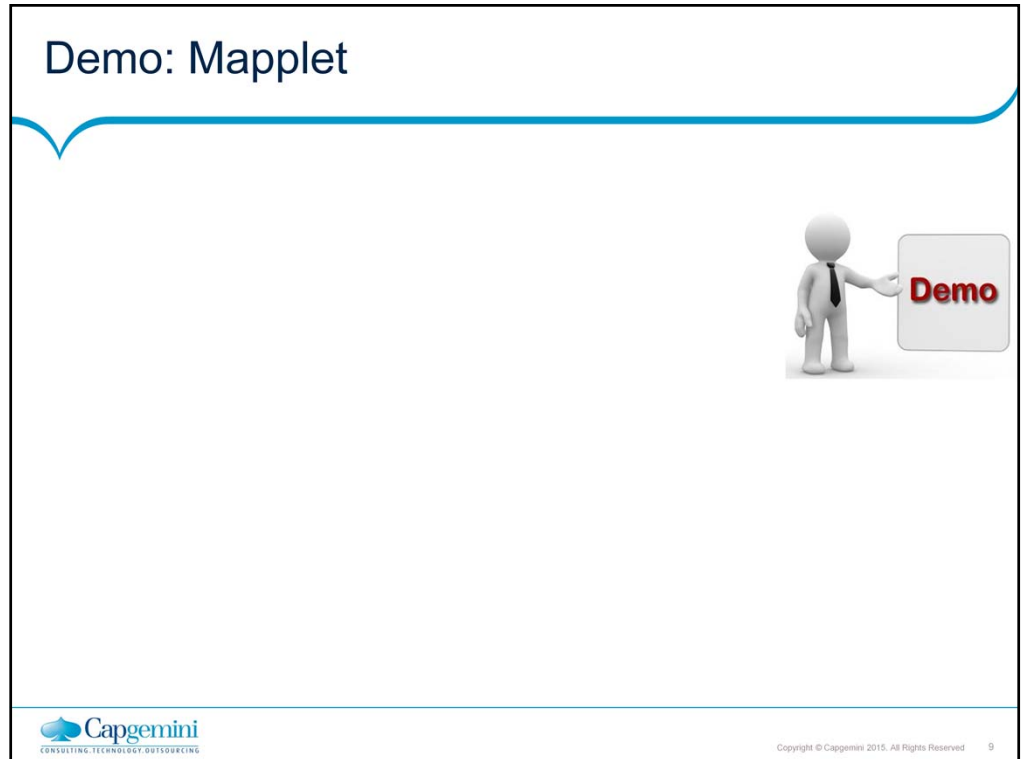
Every maplet ends with a Maplet Output Transformation. The data from the maplet passes to the mapping through the Maplet Output Transformation. A maplet must contain at least one Output transformation with at least one connected port in the maplet. Each connected port in an Output transformation displays as a maplet output port in a mapping. Each Output transformation in a maplet displays as an output group in a mapping. An output group can pass data to multiple pipelines in a mapping.

A maplet can be expanded in the Mapping Designer by selecting it and choosing Mappings-Expand from the menu. This expands the maplet within the mapping for view.

5.2. Mapplet Input and Output

Creating the Data Profile

- After you create the mapplet to filter and aggregate the data, you can profile the mapplet output data. From the Profile Manager, create a custom profile.

**Steps to create and configure a mapplet:**

1. Create a mapplet. Choose **Mapplets / Create** from the menu in the Mapplet Designer. The recommended naming convention for mapplets is *mplt_MappletName*.
2. Create Mapplet Transformation logic. Create and Link Transformations in the same manner as in a mapping.
3. Create mapplet ports.

5.3. Normalizer Transformation

Description

- Normalization is the process of organizing data
- The Normalizer Transformation normalizes records from COBOL and relational sources
- It is primarily used to create multiple rows from a single row of data
- It is used instead of the Source Qualifier Transformation for a COBOL source



Copyright © Capgemini 2015. All Rights Reserved 10

Normalization is the process of organizing data. In database terms, this includes creating normalized tables and establishing relationships between those tables according to rules designed to both protect the data and make the database more flexible by eliminating redundancy and inconsistent dependencies.

The Normalizer Transformation normalizes records from COBOL and relational sources. It allows to organize the data according to the requirements. A Normalizer transformation can appear anywhere in a data flow when a relational source is normalized.

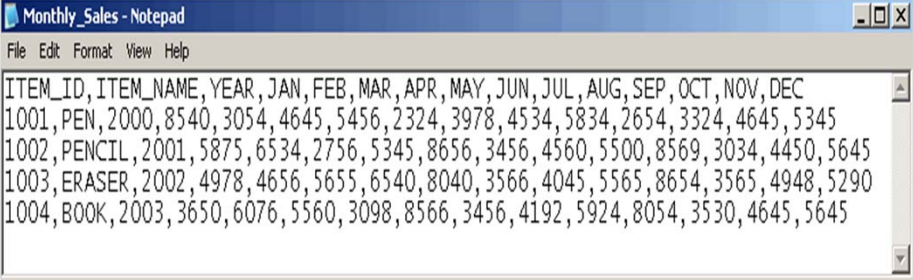
A Normalizer Transformation is used instead of the Source Qualifier Transformation when a COBOL source has to be normalized. When a COBOL source is dragged into the Mapping Designer workspace, the Normalizer Transformation automatically appears, creating input and output ports for every column in the source. COBOL sources are often stored in a denormalized format. The OCCURS statement in a COBOL file nests multiple records of information in a single record.

The Normalizer Transformation is used to break out repeated data within a record into separate records. For each new record it creates, the Normalizer Transformation generates a unique identifier. This key value is used to join the normalized records. The Normalizer Transformation can also be used with relational sources to create multiple rows from a single row of data.

5.3. Normalizer Transformation

Sample Source Data

- Item-wise monthly sales for each year



ITEM_ID	ITEM_NAME	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
1001	PEN	2000	8540	3054	4645	5456	2324	3978	4534	5834	2654	3324	4645	5345
1002	PENCIL	2001	5875	6534	2756	5345	8656	3456	4560	5500	8569	3034	4450	5645
1003	ERASER	2002	4978	4656	5655	6540	8040	3566	4045	5565	8654	3565	4948	5290
1004	BOOK	2003	3650	6076	5560	3098	8566	3456	4192	5924	8054	3530	4645	5645

Figure 5.4

In the above example, the source flat file contains the year-wise, monthly sales information in one row. The objective is to split this one row of data into multiple rows to get sales information by month.

5.3. Normalizer Transformation

Operations Performed

- A unique key called GKID is created for each new record.
- The Designer generates one column (port) for each OCCURS clause specified in the Normalizer tab to specify the positional index within an OCCURS clause.
- A column id by the name GCID_occuring_field_name is generated for each OCCURS port.



Copyright © Capgemini 2015. All Rights Reserved 12

The first step, after bringing in the source definition, would be to create a Normalizer Transformation. Ports must be manually added under the **Normalizer** tab – they cannot be dragged from other transformations. Defining the output records structure here automatically creates the input ports that are required to create the desired output. The value in the **Occurs** column of the **Normalizer** tab designates the number of repeated ports created in the Ports tab.

5.3. Normalizer Transformation

Normalizer Tab

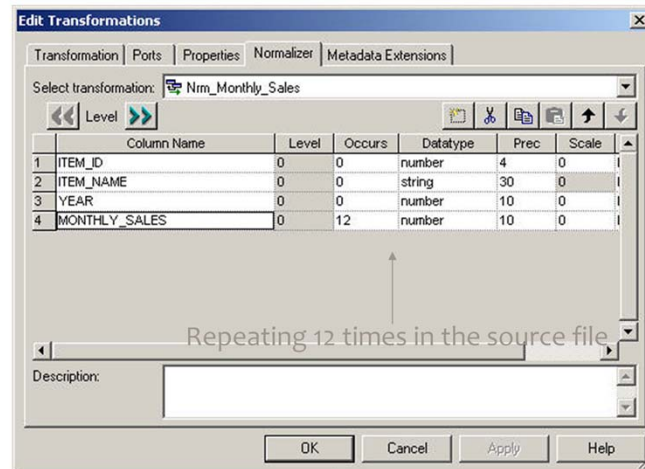


Figure 5.5

In the above example, the monthly sales will have the occurs value as 12.

5.3. Normalizer Transformation

Generated Keys

Name	Datatype
ITEM_ID_in	decimal
ITEM_NAME_in	string
YEAR_in	string
QUARTERLY_...	decimal
QUARTERLY_...	decimal
QUARTERLY_...	decimal
QUARTERLY_...	decimal
ITEM_ID	decimal
ITEM_NAME	string
YEAR	string
QUARTERLY_...	decimal
GK_QUARTE...	bigint
GCID_QUART...	integer

Figure 5.6

For each new record it creates, the Normalizer transformation generates a unique identifier called GKID_occurring_field_name. This key value can be used to join the normalized records.

The Designer also generates one column (port) for each OCCURS clause in the source file to specify the positional index within an OCCURS clause. The naming convention for the Normalizer column ID is: GCID_occurring_field_name.

5.3. Normalizer Transformation

Output

GKID_MONTHLY_SALES

ITEM_MASTER_ID	ITEM_ID	ITEM_NAME	YEAR
1	1001	PEN	2000
2	1002	PENCIL	2001
3	1003	ERASER	2002
4	1004	BOOK	2003

GCID_MONTHLY_SALES

ITEM_MASTER_ID	MONTH	MONTH_SALES
1	1	8540
1	2	3054
1	3	4645
1	4	5456
1	5	2324
1	6	3978
1	7	4534
1	8	5834
1	9	2654
1	10	3324
1	11	4645
1	12	5345

Figure 5.8

As shown in the above Figure 5.7, this generated unique key is GKID_MONTHLY_SALES and the generated column id is GCID_MONTHLY_SALES for each OCCURS in the source. The Normalizer ID columns informs about the order of records in an OCCURS clause. For example, if a record occurs two times, when the Workflow is run, the Integration service numbers the first record 1 and the second record 2. The Normalizer column ID is also useful to pivot input columns into rows.

Demo: Normalizer Transformation



Copyright © Capgemini 2015. All Rights Reserved 16

Steps to create a Normalizer Transformation:

1. In the Designer, create a new mapping or open an existing one.
2. Click and drag an imported source definition into the mapping. This source definition can be a COBOL file, a simple data file or even a relational source.
3. For a COBOL source, the Designer creates a Normalizer Transformation by default. For others, manually create the Normalizer Transformation.
4. For the COBOL source, Designer connects the Normalizer Transformation to the COBOL source definition. Open the new Normalizer Transformation.
5. For a COBOL source, select the **Ports** tab and review the ports in the Normalizer Transformation. For others, the port details will have to be entered in the Normalizer tab.
6. Select the **Normalizer** tab and add new output ports. Add a port corresponding to each column in the source record that contains denormalized data. The new ports only allow the number or string datatypes. Only new ports can be created in the **Normalizer** tab.
7. Set the appropriate OCCURS value for the specific source data column.

Summary

- After completing this lesson you now:
 - Know how to configure a Mapplet using:
 - Mapplet Input Transformation
 - Mapplet Output Transformation
 - Know how to configure a Normalizer Transformation







Review Question

- Question 1: Mapplets are configured using the _____.
- Question 2: A mapplet cannot have _____ definitions.
- Question 3: A Normalizer Transformation can be used for COBOL, VSAM and relational sources.
 - True/False
- Question 4: GKID is a unique key generated for each column with the OCCURS clause.
 - True/False

