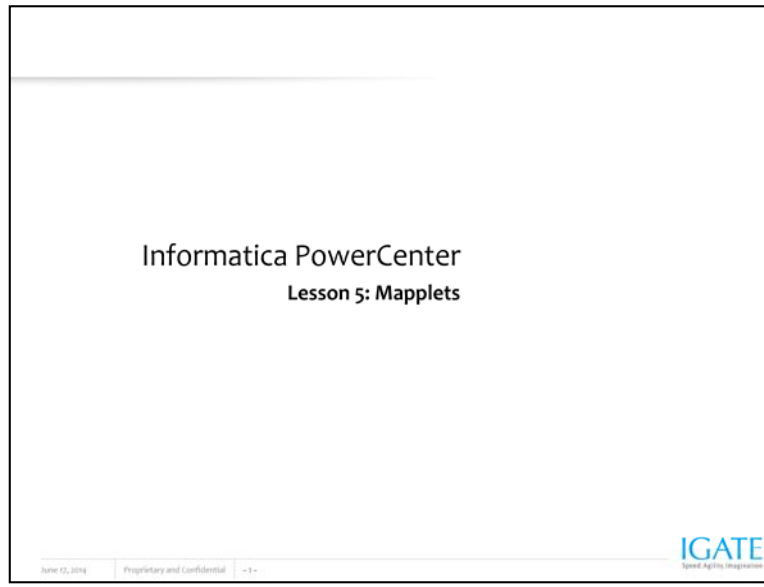
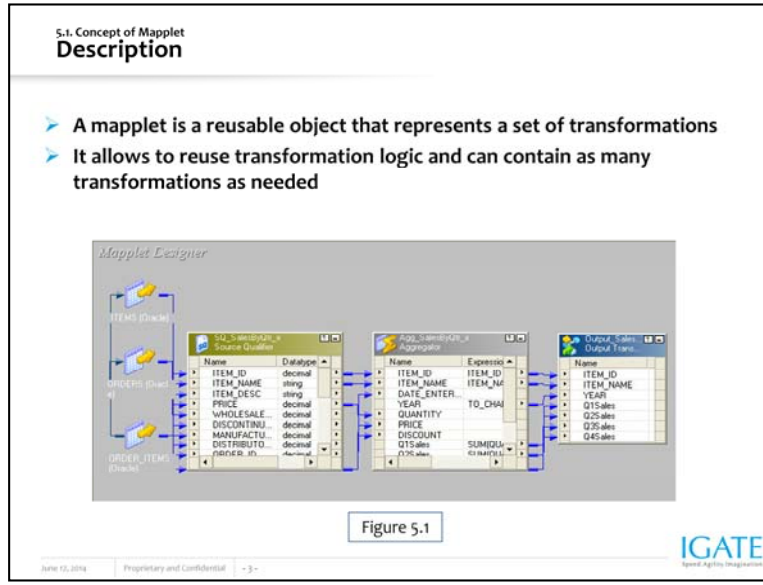
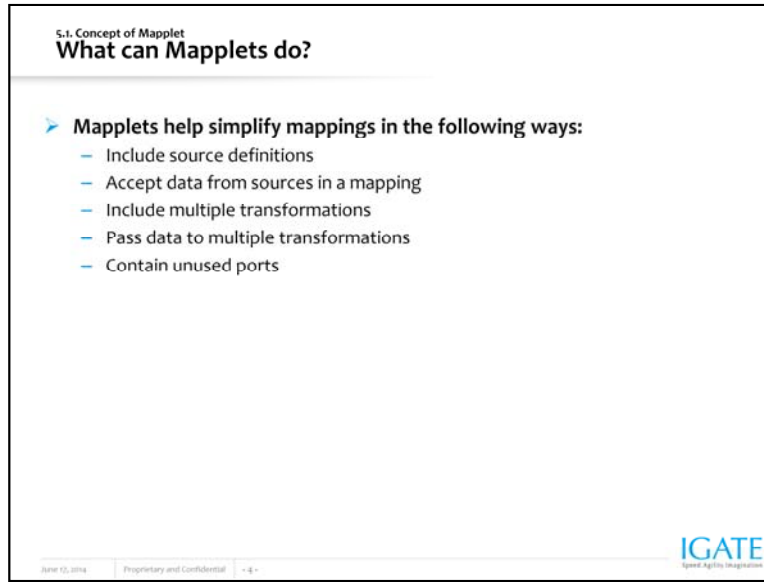

Informatica PowerCenter
Lesson 5: Mapplets
IGATE

Lesson Objectives

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

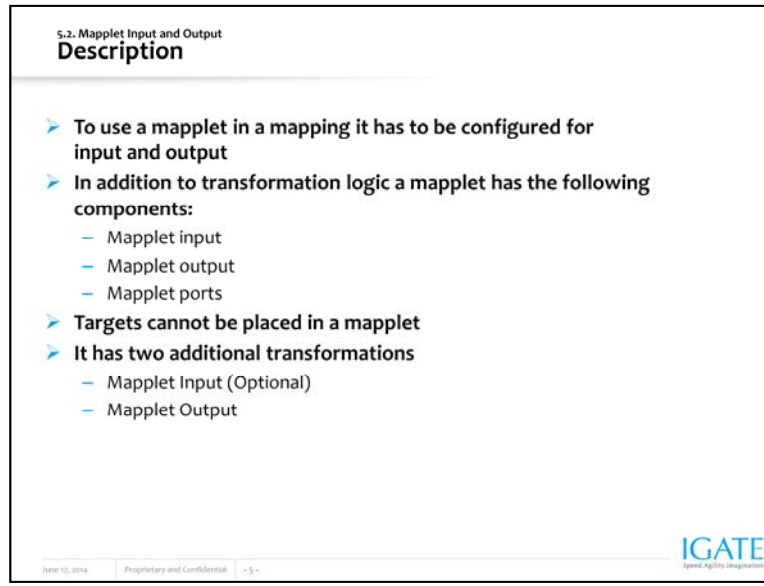


A *mapplet* is a reusable object that is created in the Mapplet Designer. It contains a set of transformations. It allows to reuse the transformation logic in multiple mappings.



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



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.

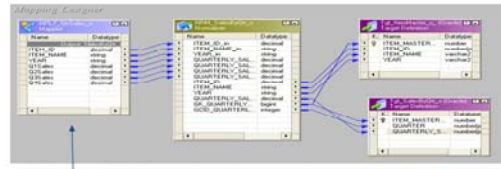
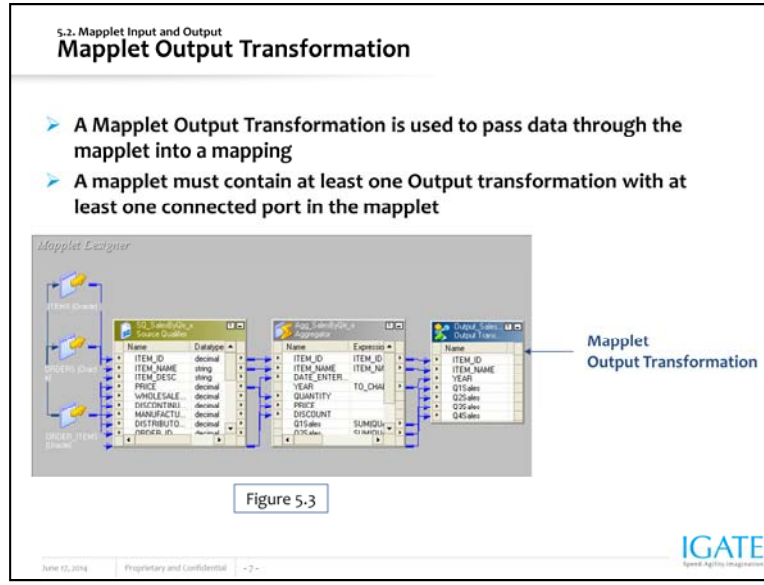


Figure 5.2

Mapplet Input Transformation

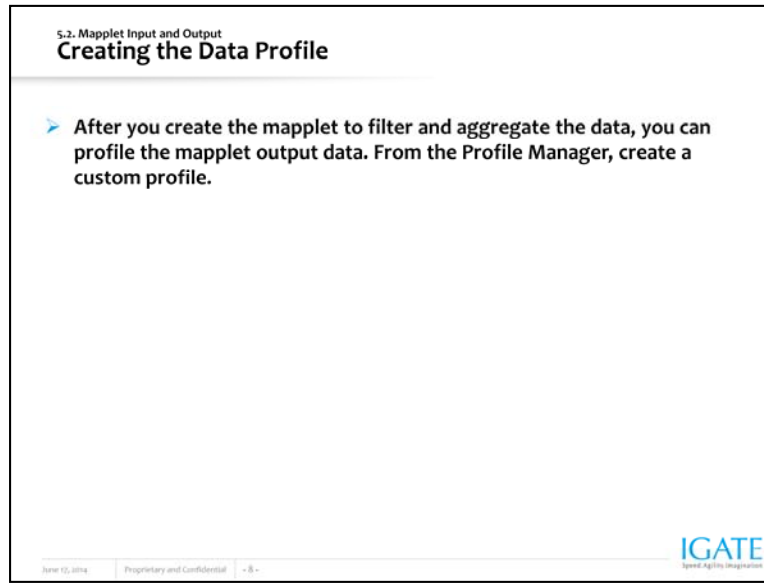
IGATE

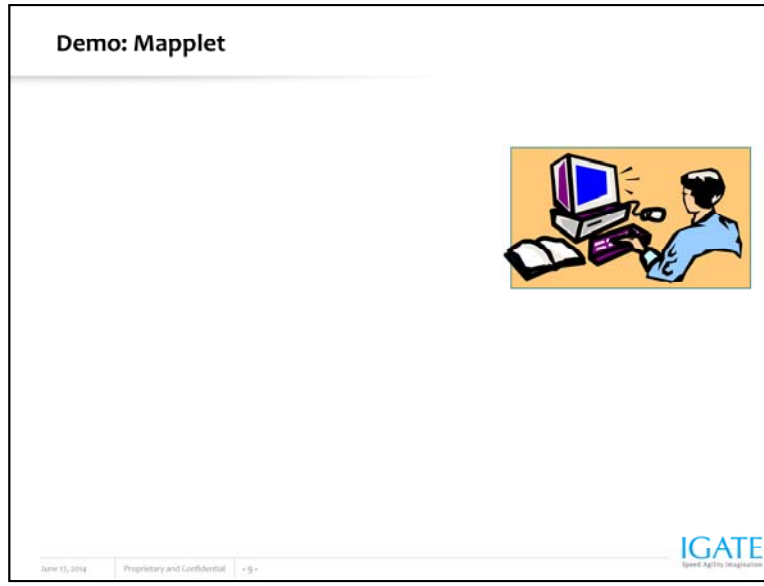
An Input Transformation is used to receive input from a source in a mapping. When the mapplet is used in a mapping, the Input transformation provides input ports so data can be passed through the mapplet. Each port in the Input transformation connected to another transformation in the mapplet becomes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



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

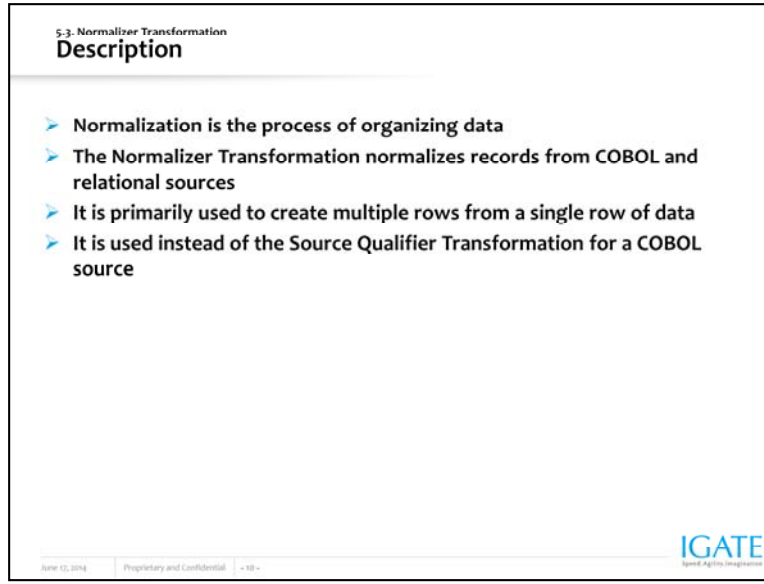
A mapplet can be expanded in the Mapping Designer by selecting it and choosing Mappings-Expand from the menu. This expands the mapplet within the mapping for view.


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.
IGATE



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.

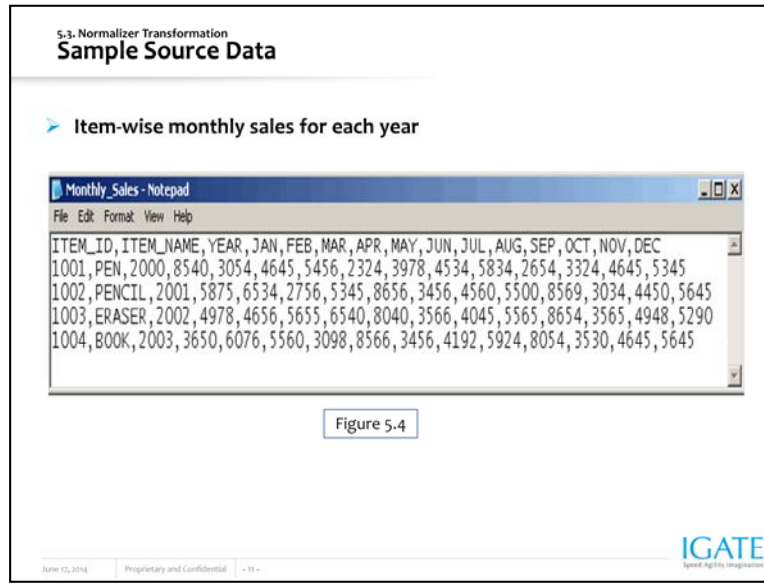


Normalization is the process of organizing data. In database terms, this includes creating normalized tables and establishing relationships between those tables according to rules designed to both protect the data and make the database more flexible by eliminating redundancy and inconsistent dependencies.

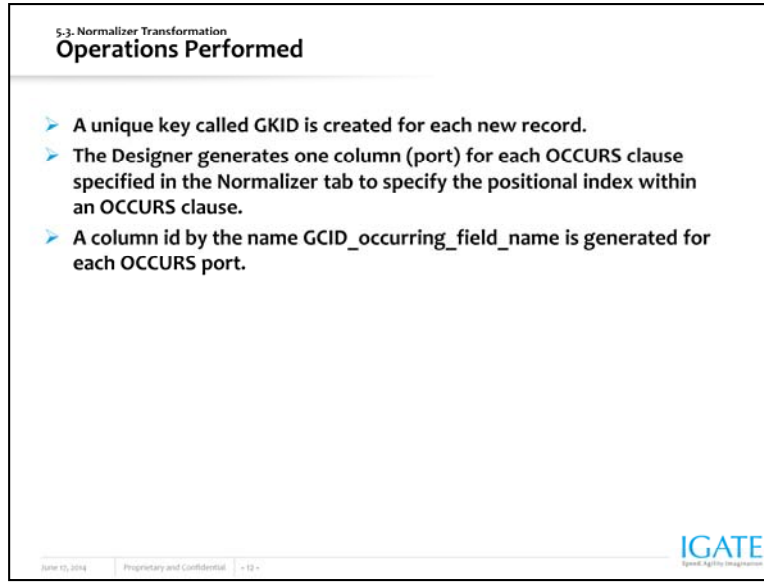
The *Normalizer Transformation* normalizes records from COBOL and relational sources. It allows to organize the data according to the requirements. A Normalizer transformation can appear anywhere in a data flow when a relational source is normalized.

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.

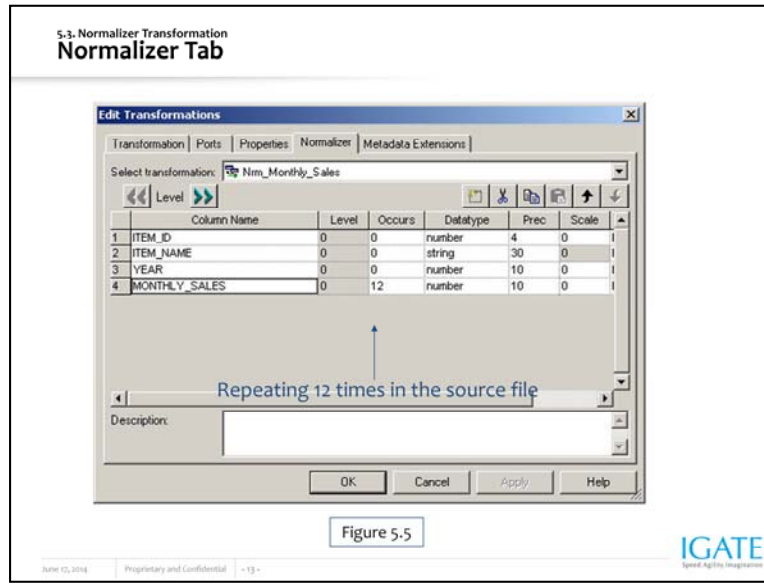
The *Normalizer Transformation* is used to break out repeated data within a record into separate records. For each new record it creates, the Normalizer Transformation generates a unique identifier. This key value is used to join the normalized records. The Normalizer Transformation can also be used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



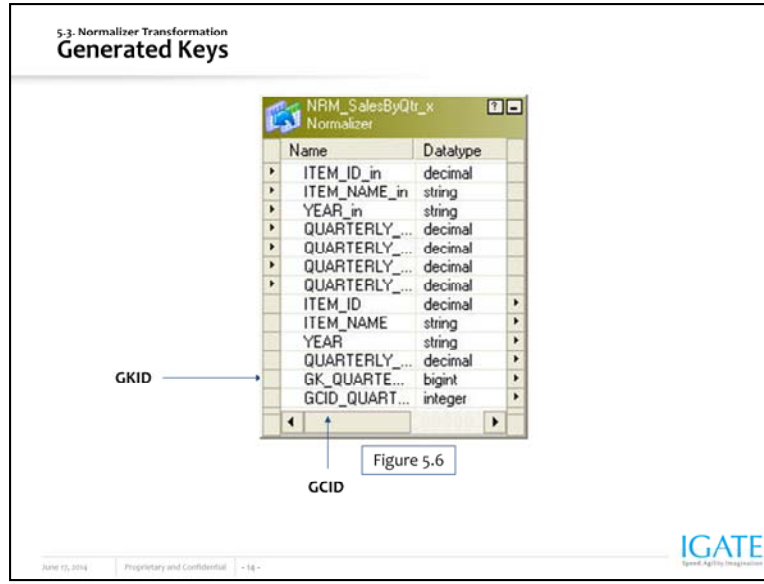
In the above example, the source flat file contains the year-wise, monthly sales information in one row. The objective is to split this one row of data into multiple rows to get sales information by month.



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.

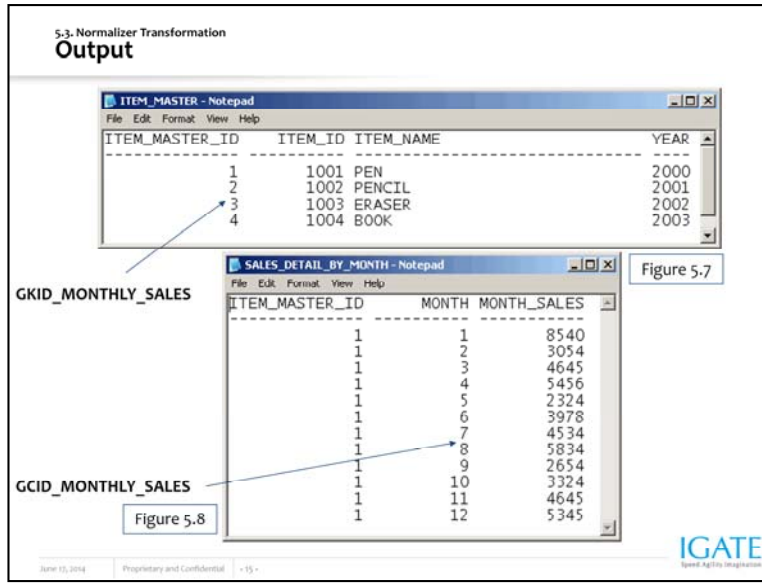


In the above example, the monthly sales will have the occurs value as 12.

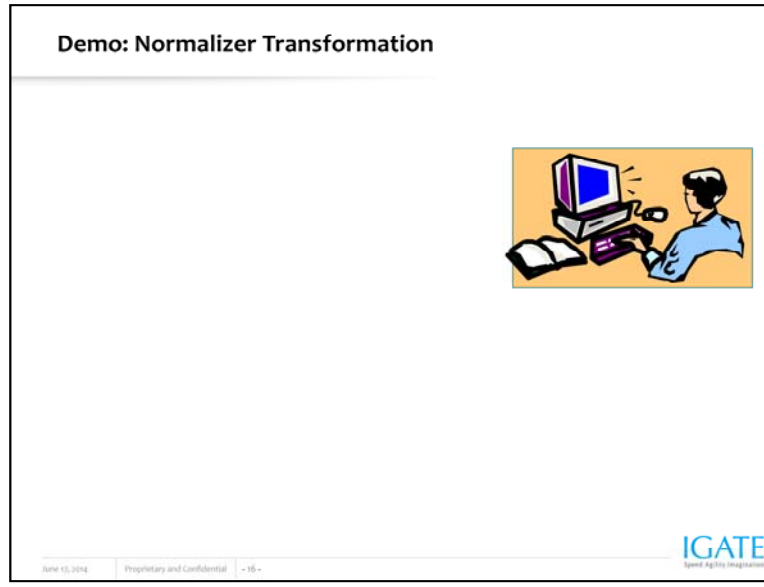


For each new record it creates, the Normalizer transformation generates a unique identifier called GKID_occurring_field_name. This key value can be used to join the normalized records.

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.



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.

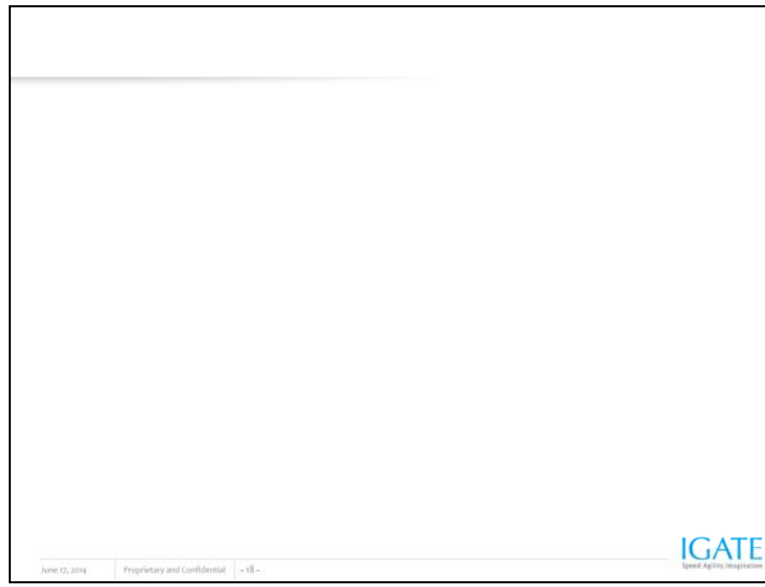


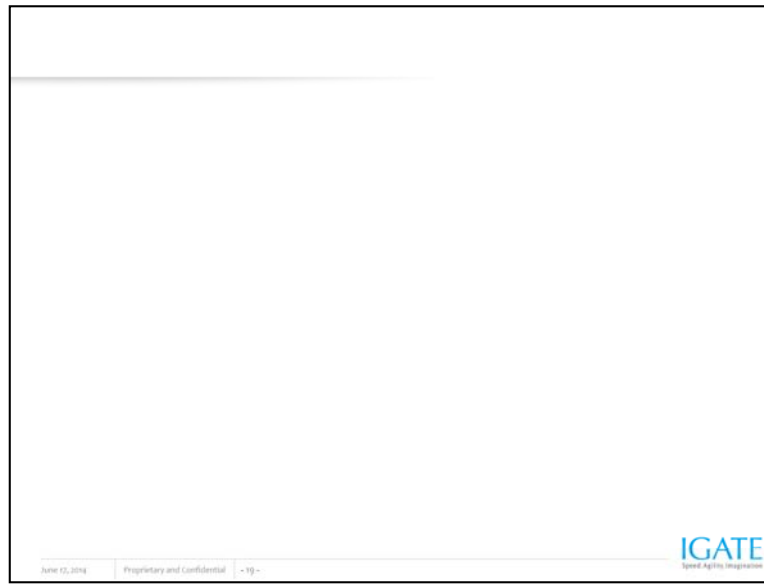
Steps to create a Normalizer Transformation:

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 Trasformation
  - Know how to configure a Normalizer Transformation


June 12, 2014
Proprietary and Confidential
- 18 -
IGATE


June 17, 2014
Proprietary and Confidential
- 19 -
IGATE

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