**Scenarios- :**

We can create the Data validations, Special characters removal, Junk characters removal and Lookup transformations functionalities inside a mapplet and we can call the Mapplet in our Informatica mappings for reusable purposes.

1.Length Check validation

2.Junk Characters removal

3. Handling Special characters

4. Mapplet-(Lookup)

**Mapplets**

A mapplet is a reusable object that we can create in the Mapplet Designer. It contains a set of transformations and lets you reuse the transformation logic in multiple mappings.

For example, if you have several fact tables that require a series of dimension keys, you can create a mapplet containing a series of Lookup transformations to find each dimension key. You can then use the mapplet in each fact table mapping, rather than recreate the same lookup logic in each mapping.

When you use a mapplet in a mapping, you use an instance of the mapplet. Like a reusable transformation, any change made to the mapplet is inherited by all instances of the mapplet.

Mapplets help simplify mappings in the following ways:

* **Include source definitions.** Use multiple source definitions and source qualifiers to provide source data for a mapping.
* **Accept data from sources in a mapping.** If you want the mapplet to receive data from the mapping, use an Input transformation to receive source data.
* **Include multiple transformations.** A mapplet can contain as many transformations as you need.
* **Pass data to multiple transformations.** You can create a mapplet to feed data to multiple transformations. Each Output transformation in a mapplet represents one output group in a mapplet.
* **Contain unused ports.** You do not have to connect all mapplet input and output ports in a mapping.

To use a mapplet in a mapping, you must configure it for input and output. In addition to transformation logic that you configure, a mapplet has the following components:

* **Mapplet input.** You can pass data into a mapplet using source definitions or Input transformations or both. When you use an Input transformation, you connect it 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. If a mapplet uses source definitions rather than Input transformations for input, it does not contain any input ports in the mapping.

## Mapplet Input

Mapplet input can originate from a source definition and/or from an Input transformation in the mapplet. You can create multiple pipelines in a mapplet. Use multiple source definitions and source qualifiers or Input transformations. You can also use a combination of source definitions and Input transformations.

### Using Source Definitions for Mapplet Input

Use one or more source definitions in a mapplet to provide source data. When you use the mapplet in a mapping, it is the first object in the mapping pipeline and contains no input ports.

### Using Input Transformations for Mapplet Input

Use an Input transformation in a mapplet when you want the mapplet to receive input from a source in a mapping. When you use the mapplet in a mapping, the Input transformation provides input ports so you can pass data 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. Unconnected ports do not display in the Mapping Designer.

You can connect an Input transformation to multiple transformations in a mapplet. However, you cannot connect a single port in the Input transformation to multiple transformations in the mapplet.

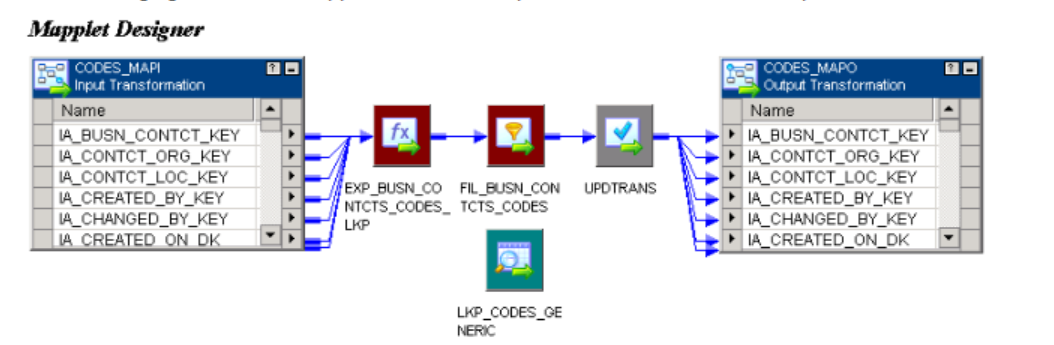
## Mapplet Output

Use an Output transformation in a mapplet to pass data through the mapplet into a mapping. 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.

## Viewing Mapplet Input and Output

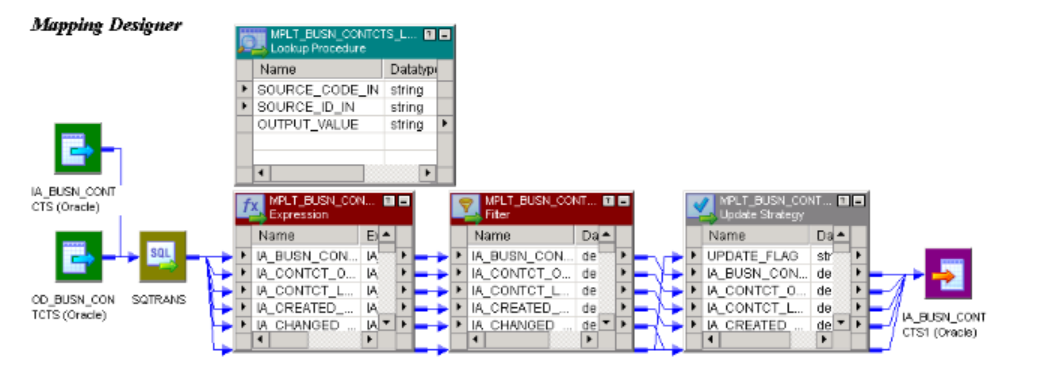
Mapplets and mapplet ports display differently in the Mapplet Designer and the Mapping Designer.

The following figure shows a mapplet with both an Input transformation and an Output transformation



You can open or iconize all the transformations in the mapplet and mapping. You cannot edit any of the properties, navigate to other folders, or save the repository while the mapplet is expanded.

The following figure shows an expanded mapplet in the Mapping Designer:



In an expanded mapping, you do not see the Input and Output transformations.

# Using the Mapplet Designer

After you create a mapplet, you can validate or edit the mapplet in the Mapplet Designer. You can also use the Designer to copy mapplets, export and import mapplets, view links between ports in a mapplet, create shortcuts to mapplets, and delete mapplets from the repository.

To create and configure a mapplet in the Mapplet Designer, complete the following steps:

1. Create a mapplet. Click 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.

## Creating a Mapplet

A mapplet can be active or passive depending on the transformations in the mapplet. Active mapplets contain one or more active transformations. Passive mapplets contain only passive transformations. When you use a mapplet in a mapping, all transformation rules apply to the mapplet depending on the mapplet type. For example, as with an active transformation, you cannot concatenate data from an active mapplet with a different pipeline.

Use the following rules and guidelines when you add transformations to a mapplet:

* If you use a Sequence Generator transformation, you must use a reusable Sequence Generator transformation.
* If you use a Stored Procedure transformation, you must configure the Stored Procedure Type to be Normal.
* You cannot include PowerMart 3.5-style LOOKUP functions in a mapplet.
* You cannot include the following objects in a mapplet:
  + Normalizer transformations
  + COBOL sources
  + XML Source Qualifier transformations
  + XML sources
  + Target definitions
  + Other mapplets

Although reusable transformations and shortcuts in a mapplet can be used, to protect the validity of the mapplet, use a copy of a transformation instead. Reusable transformations and shortcuts inherit changes to their original transformations. This might invalidate the mapplet and the mappings that use the mapplet.

## Validating Mapplets

The Designer validates a mapplet when you save it. You can also validate a mapplet using the Mapplets > Validate menu command. When you validate a mapplet, the Designer writes all relevant messages about the mapplet in the Output window.

The Designer validates the mapplet pipeline in the same way it validates a mapping. The Designer also performs the following checks specific to mapplets:

* The mapplet can contain Input transformations and source definitions with at least one port connected to a transformation in the mapplet.
* The mapplet contains at least one Output transformation with at least one port connected to a transformation in the mapplet.

## Editing Mapplets

You can edit a mapplet in the Mapplet Designer. The Designer validates the changes when you save the mapplet. When you save changes to a mapplet, all instances of the mapplet and all shortcuts to the mapplet inherit the changes. These changes might invalidate mappings that use the mapplet.

To see what mappings or shortcuts may be affected by changes you make to a mapplet, select the mapplet in the Navigator, right-click, and select Dependencies. Or, click Mapplets > Dependencies from the menu.

You can make the following changes to a mapplet without affecting the validity of existing mappings and sessions:

* Add input or output ports.
* Change port names or comments.
* Change Input or Output transformation names or comments.
* Change transformation names, comments, or properties.
* Change port default values for transformations in the mapplet.
* Add or remove transformations in the mapplet, providing you do not change the mapplet type from active to passive or from passive to active.

Use the following rules and guidelines when you edit a mapplet that is used by mappings:

* **Do not delete a port from the mapplet.** The Designer deletes mapplet ports in the mapping when you delete links to an Input or Output transformation or when you delete ports connected to an Input or Output transformation.
* **Do not change the datatype, precision, or scale of a mapplet port.** The datatype, precision, and scale of a mapplet port is defined by the transformation port to which it is connected in the mapplet. Therefore, if you edit a mapplet to change the datatype, precision, or scale of a port connected to a port in an Input or Output transformation, you change the mapplet port.
* **Do not change the mapplet type.** If you remove all active transformations from an active mapplet, the mapplet becomes passive. If you add an active transformation to a passive mapplet, the mapplet becomes active.

# Using Mapplets in Mappings

In a mapping, a mapplet has input and output ports that you can connect to other transformations in the mapping. You do not have to connect all mapplet ports in a mapping. However, if the mapplet contains an SQL override, you must connect all mapplet output ports in the mapping.

Like a reusable transformation, when you drag a mapplet into a mapping, the Designer creates an instance of the mapplet. You can enter comments for the instance of the mapplet in the mapping. You cannot otherwise edit the mapplet in the Mapping Designer.

If you edit the mapplet in the Mapplet Designer, each instance of the mapplet inherits the changes.

The PowerCenter Repository Reports has a Mapplets list report that you use to view all mappings using a particular mapplet.

To use a mapplet, complete the following steps:

1. Drag the mapplet into the mapping.
2. If the mapplet contains input ports, connect at least one mapplet input port to a transformation in the mapping.
3. Connect at least one mapplet output port to a transformation in the mapping.

## Creating and Configuring Mapplet Ports

After creating transformation logic for a mapplet, you can create mapplet ports. Use an Input transformation to define mapplet input ports if the mapplet contains no source definitions. Use an Output transformation to create a group of output ports. Only connected ports in an Input or Output transformation become mapplet input or output ports in a mapping. Unconnected ports do not display when you use the mapplet in a mapping.

You can create a mapplet port in the following ways:

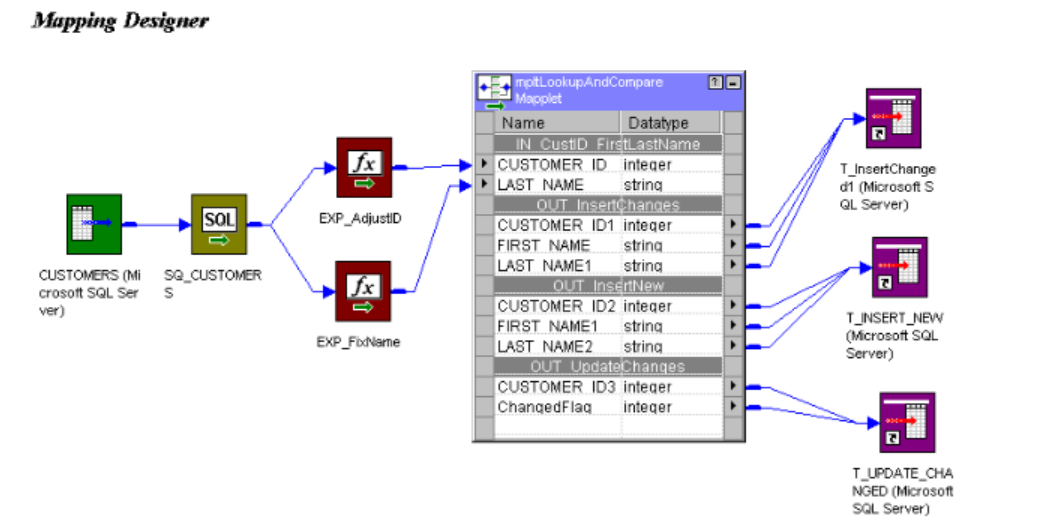
* **Manually create ports in the Input/Output transformation.** You can create port names in Input and Output transformations. You can also enter a description for each port name. The port has no defined datatype, precision, or scale until you connect it to a transformation in the mapplet.
* **Drag a port from another transformation.** You can create an input or output port by dragging a port from another transformation into the Input or Output transformation. The new port inherits the port name, description, datatype, and scale of the original port. You can edit the new port name and description in the transformation. If you change a port connection, the Designer updates the Input or Output transformation port to match the attributes of the new connection.

You can view the datatype, precision, and scale of available mapplet ports when you use the mapplet in a mapping.

## Connecting to Mapplet Input Ports

When using a mapplet with input ports in a mapping, you connect the mapplet input ports to the mapping pipeline. You can pass data into a mapplet only when it originates from a single active transformation.

For example, in the following figure, the mapplet mpltLookupAndCompare accepts data from two Expression transformations because data from both transformations originate from a single source qualifier. The Source Qualifier SQ\_CUSTOMERS is the active transformation providing mapplet source data:



## Connecting to Mapplet Output Groups

Each Output transformation displays as an output group when you use a mapplet in a mapping. Connect the mapplet output ports to the mapping pipeline. Use Autolink to connect the ports.

Use the following rules and guidelines when you connect mapplet output ports in the mapping:

* When a mapplet contains a source qualifier that has an override for the default SQL query, you must connect all of the source qualifier output ports to the next transformation within the mapplet.
* If the mapplet contains more than one source qualifier, use a Joiner transformation to join the output into one pipeline.
* If the mapplet contains only one source qualifier, you must connect the mapplet output ports to separate pipelines. You cannot use a Joiner transformation to join the output.

If you need to join the pipelines, you can create two mappings to perform this task:

* + Use the mapplet in the first mapping and write data in each pipeline to separate targets.
  + Use the targets as sources in the second mapping to join data, then perform any additional transformation necessary.

**Data Validations using Informatica transformations**

Informatica transformations create, modify, or pass data to a defined target structure (tables, files, or other targets). Their two main goals are:

* They modify the source data as per the target system’s requirements.
* They ensure the loading of data quality into the target.

**Types of transformation**

There are two categories of transformation.

**Based on connectivity**

* Connected Transformations: You can use connected transformations when you get a transformation request for every input row or when the value is expected to return.
* Unconnected Transformations: The unconnected transformations are only useful periodically or based upon certain conditions.

**Based on the change in the number of rows**

* Active Transformations: Active transformations modify the data rows and the number of input rows.
* Passive Transformations: Passive transformations do not change the number of input rows.

**Informatica Substring function**

The substring function retrieves a particular piece of the string. Substring counts blank characters in the string. For example,

Substring( Australia, 4, 7)

“4” is the position of the starting character of the substring.

“8” is the position of the ending character of the substring.

**Informatica Not NULL**

You can do the Not NULL check by using the Informatica ISNULL function. There are two ways to do this:

1. Using the NOT Operator

IIF(NOT ISNULL(value), ' Input is not null',' No it is null')

1. Checking with the return value.

IIF(ISNULL(value)=0, ' Input is not null',' No it is null')

Verbose initialization mode in Informatica

In the verbose initialization mode, Informatica writes the log details as same as normal mode. It also writes about the data and index files used and transformation statistics.

Regular Expression in Informatica

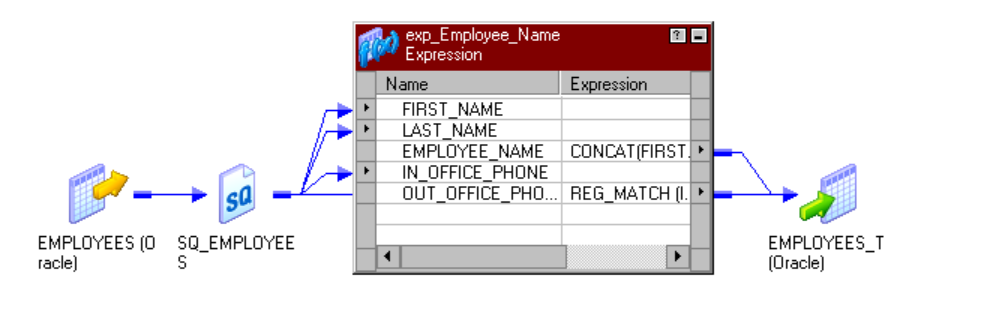
In Informatica, a regular expression includes characters representing the source character types, source character sets, and string or word boundaries in the source columns. It can also include quantifiers that decide how many times a character can occur in the source data. Regular expressions are case sensitive.

**Expression Transformation**

Use the Expression transformation to calculate values in a single row. For example, you might need to adjust employee salaries, concatenate first and last names, or convert strings to numbers. You can also use the Expression transformation to test conditional statements before you pass the results to a target or other transformations. The Expression transformation is a passive transformation.

Use the Expression transformation to perform non-aggregate calculations. To perform calculations involving multiple rows, such as sums or averages, use the Aggregator transformation.

The following figure shows a simple mapping with an Expression transformation used to concatenate the first and last names of employees from the EMPLOYEES table:

Creating an Expression Transformation

Use the following procedure to create an Expression transformation:

1. In the Mapping Designer, open a mapping.
2. Click Transformation > Create. Select Expression transformation.
3. Enter a name and click Done.
4. Select and drag the ports from the source qualifier or other transformations to add to the Expression transformation.

You can also open the transformation and create ports manually.

1. Double-click on the title bar and click on Ports tab. You can create output and variable ports within the transformation.
2. Assign the port datatype, precision, and scale to match the expression return value.
3. In the Expression section of an output or variable port, open the Expression Editor.
4. Enter an expression.
5. Click Validate to verify the expression syntax.
6. To reflect the latest changes from the expression condition in the test expression section in the right pane, click Refresh.
7. In the right pane, enter sample values for the input ports used within the expression.
8. To evaluate the expression, click Evaluate.
9. Click OK.
10. Create reusable transformations on the Transformation tab.

Note: After you make the transformation reusable, you cannot copy ports from the source qualifier or other transformations. You can create ports manually within the transformation.

1. Configure the tracing level on the Properties tab.
2. Add metadata extensions on the Metadata Extensions tab.
3. Click OK.

Connect the output ports to a downstream transformation or target

The following functions used for the special characters removal purpose inside an expression transformation.

1. REG\_REPLACE(PORT\_NAME, '[^\w,-. ]', '')

This function removes the special characters and retains only alphanumeric characters, commas, dashes, and periods.

2. REG\_REPLACE(PORT\_NAME,'[^a-z0-9A-Z]','')

This function retains alphanumeric characters only.

3. REG\_REPLACE(PORT\_NAME,'[^[:print:]]','')

This function looks for '[^[:print:]]' which means its searching for non printable characters in the field which we are passing and its been replaced with '' (NULL).

For Example an username column can be checked by using the below syntax

REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(username,'Ü','UE'),'Ö','OE'),'ß','ss'),'Ä','AE'),'ä','ae'),'ü','ue'),'ö','oe') as user\_name

**Lookup transformation**

Use a Lookup transformation in a mapping to look up data in a flat file, relational table, view, or synonym. You can import a lookup definition from any flat file or relational database to which both the PowerCenter Client and Integration Service can connect. You can also create a lookup definition from a source qualifier. You can use multiple Lookup transformations in a mapping. The Lookup transformation can be an active or passive transformation. You can configure a connected or unconnected Lookup transformation.

The Integration Service queries the lookup source based on the lookup ports in the transformation and a lookup condition. The Lookup transformation returns the result of the lookup to the target or another transformation. You can configure the Lookup transformation to return a single row or multiple rows.

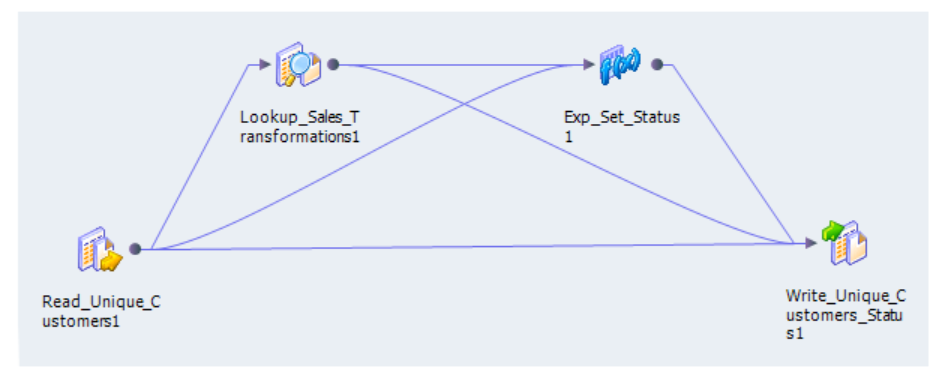
Perform the following tasks with a Lookup transformation:

* **Get a related value.** Retrieve a value from the lookup table based on a value in the source. For example, the source has an employee ID. Retrieve the employee name from the lookup table.
* **Get multiple values**. Retrieve multiple rows from a lookup table. For example, return all employees in a department.
* **Perform a calculation.** Retrieve a value from a lookup table and use it in a calculation. For example, retrieve a sales tax percentage, calculate a tax, and return the tax to a target.
* **Update slowly changing dimension tables.** Determine whether rows exist in a target.

Configure the Lookup transformation to perform the following types of lookups:

* **Relational or flat file lookup.** Perform a lookup on a flat file or a relational table. When you create a Lookup transformation using a relational table as the lookup source, you can connect to the lookup source using ODBC and import the table definition as the structure for the Lookup transformation. When you create a Lookup transformation using a flat file as a lookup source, the Designer invokes the Flat File Wizard.
* **Pipeline lookup.** Perform a lookup on application sources such as JMS or MSMQ. Drag the source into the mapping and associate the Lookup transformation with the source qualifier. Configure partitions to improve performance when the Integration Service retrieves source data for the lookup cache.
* **Connected or unconnected lookup.** A connected Lookup transformation receives source data, performs a lookup, and returns data to the pipeline. An unconnected Lookup transformation is not connected to a source or target. A transformation in the pipeline calls the Lookup transformation with a :LKP expression. The unconnected Lookup transformation returns one column to the calling transformation.
* **Cached or uncached lookup.** Cache the lookup source to improve performance. If you cache the lookup source, you can use a dynamic or static cache. By default, the lookup cache remains static and does not change during the session. With a dynamic cache, the Integration Service inserts or updates rows in the cache. When you cache the target table as the lookup source, you can look up values in the cache to determine if the values exist in the target. The Lookup transformation marks rows to insert or update the target.

A connected Lookup transformation is a Lookup transformation that is connected to a source or target in a mapping. The following figure shows a mapping with a connected Lookup transformation:



# Creating a Lookup Transformation

Create a reusable Lookup transformation in the Transformation Developer. Create a non-reusable Lookup transformation in the Mapping Designer.

To create a Lookup transformation:

1. To create a reusable Lookup transformation, open the Transformation Developer. To create a non-reusable Lookup transformation, open a mapping in the Mapping Designer. If you are creating pipeline Lookup transformation, drag in a source definition to use as a lookup source.
2. Click Transformation > Create. Select the Lookup transformation.
3. Enter a name for the transformation. Click Create.

The naming convention for Lookup transformations is LKP\_TransformationName.

1. Choose whether the transformation is active or passive. Click OK. You cannot change this option.
2. In the Select Lookup Table dialog box, choose one of the following options to import a lookup definition:
   * Source definition from the repository.
   * Target definition from the repository.
   * Source qualifier from the mapping.
   * Import a relational table or file from the repository.

Note: You can manually add the lookup ports instead of importing a definition. You can choose which lookup ports are also output ports.

When you choose the lookup source, the Designer creates ports in the transformation based on the ports in the object that you choose. The Designer configures each port as a lookup port and an output port. The lookup ports represent the columns in the lookup source. The Lookup transformation receives data from the lookup source in each lookup port and passes the data to the target.

1. If you want the Lookup transformation to return all matching rows, enable Return All Rows on Multiple Match. You cannot change this option after you create the transformation. The Lookup transformation becomes an active transformation.
2. Click OK or click Skip if you want to manually add the lookup ports instead of importing a definition. You can choose which lookup ports are also output ports.
3. For a connected Lookup transformation, add input and output ports.

You can pass data through the transformation and return data from the lookup table to the target.

1. For an unconnected Lookup transformation, create a return port for the value you want to return from the lookup.

You can return one column to the transformation that called the lookup.

1. Click the Properties tab to configure the Lookup transformation properties. Configure lookup caching.

Lookup caching is enabled by default for pipeline and flat file Lookup transformations.

1. For a Lookup transformation that has a dynamic lookup cache, associate an input port, output port, or sequence ID with each lookup port.

The Integration Service inserts or updates rows in the lookup cache with the data from each associated expression. If you associate a sequence ID, the Integration Service generates a primary key for inserted rows in the lookup cache.

1. Add the lookup condition on the Condition tab.

The lookup condition compares the source column values with values in the lookup source. The Condition tab Transformation Port represents the source column values. The Lookup Table represents the lookup source