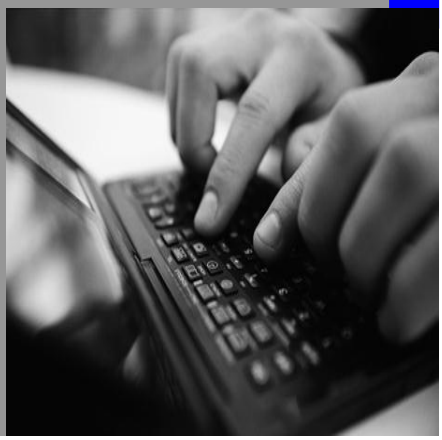


How-to Guide

Manufacturing Integration and Intelligence

# MII Manufacturing Data Objects (MDO) Guide

Version 1.0 – Sept 2012

Applicable Releases:

SAP MII 14.0

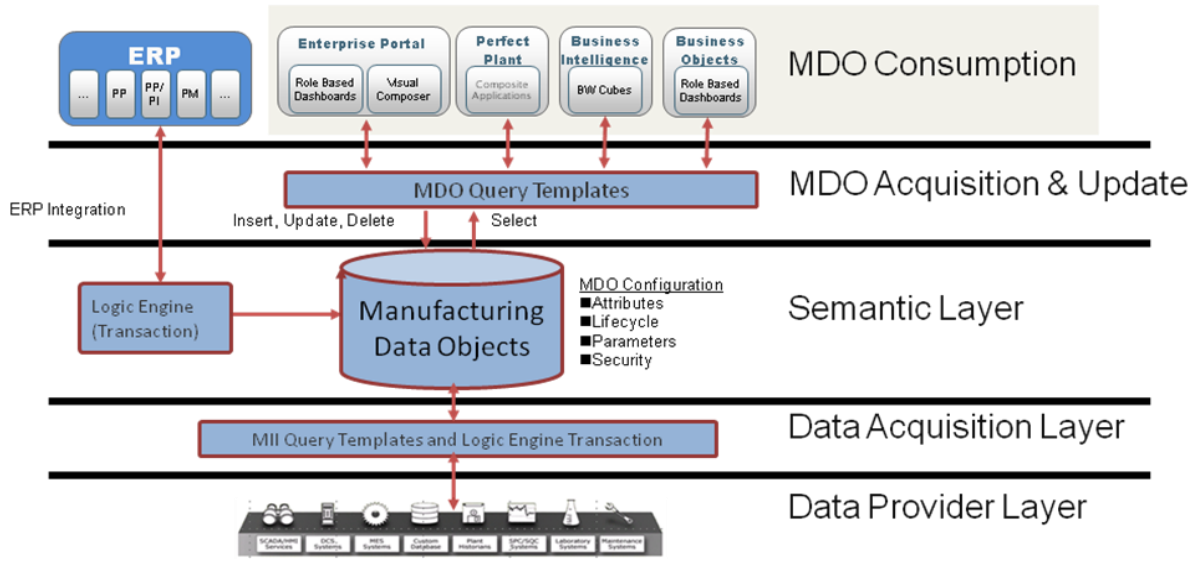
SAP NetWeaver Composition 7.30 SPX

# 1. Manufacturing Data Object (MDO)

A Manufacturing Data Object (MDO) is a semantic layer and level of abstraction on top of SAP Manufacturing Integration and Intelligence (SAP MII) data acquisition mechanisms which is used to define object models and data structures and to also persist data.

**Benefit**: Ability to model a common semantic layer utilized across applications and plants eliminating custom database development and facilitating standardized integration

The below figure depicts the overall High Level Design of the Role of *MDO* and *MDO Query Template* in SAP MII.




The Manufacturing Data Object (MDO) has now been enhanced to be OpenSQL compliant .

**Why OpenSQL?**

Open SQL allows you to access all database tables known to the SAP system where as the Native SQl allows only database-specific SQL statements. For more information on the OpenSQl Grammer :

http://help.sap.com/saphelp_nw70/helpdata/en/9b/f46cabaa874bc9a82234e8cf1d0696/content.htm

All programming models provided by SAP for database access are based on the Open SQL for Java framework. This provides performance-enhancing mechanisms, such as table buffering and statement pooling, while providing vendor independent access to various databases, such as Oracle, IBM DB2, Microsoft SQL Server, and the SAP-certified open source database MaxDB. Applications do not need to be changed because Open SQL works in all of the supported databases.

The Open SQL for Java framework provides a wrapper for the proprietary JDBC drivers. In doing so, it provides the binding layer between AS Java and proprietary JDBC drivers at runtime. This ensures efficient and portable SQL-based data selection and manipulation. With Open SQL, data in database tables, which is defined in the Java Dictionary, can be read and changed. Open SQL statements are checked fully by the syntax check.

**Why Not Native SQL?**

Native SQL are statements that can be listed in the programs between special delimiters. Database-specific SQL statements and special SAP-specific statements are possible. Native SQL statements are not checked fully by the syntax check and are handled by the Native SQL interface of the database interface.

Hence the Java program containing database-specific SQL statements will **not** run under different database systems. If your program will be used on more than one database platform, only use Open SQL statements.

**Number of Tables supported in OpenSQL**

To ensure program portability, specify no more than 15 tables in a query(including the tables in the FROM list and tables in subqueries).Individual host database management systems may allow more than 15 tables.

When you create a new MDO, you can configure it as one of the following:

- *Persistent MDO*

  The Persistent mode was designed to stage data in the predefined MDO data source for local reporting or for asynchronous transactional integration with other systems. The content stored in these objects can be queried and then reported from and is ideal for local survivability scenarios. In many cases, where custom database tables were previously required for the development of a composite application, you can now use the modeling and persistence layer available in MDO.

  When the data provider for the object is executed, the system persists the data that is returned.

- *On-Demand MDO*

  An On-Demand object is merely a pointer to a run time query. The MDO representation of the run time query provides a friendlier namespace to the UI developer, but the result is still a query result set.

  A typical use case would be to create a UI front-end whose data can be "wired-in" later. In this scenario, the on demand object acts as an abstraction layer between the UI and the underlying data source.

  When the data provider for the object is executed, the data is not persisted. Instead, data returned from the data source is presented in real-time.

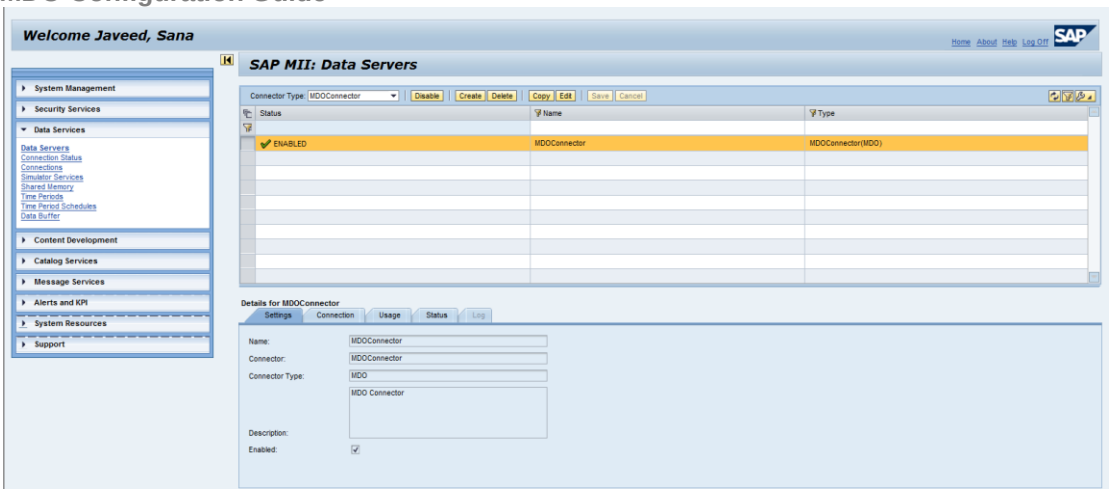
- *Joined MDO*

  Prior to MII 14.0 , data could not be obtained based on a join result on two or more Persistent MDOs. MII 14.0 has introduced a new MDO object which performs exactly that.

  Here the data provider comprises of two or more Persistent MDOs on which joins can be performed namely Inner Join and Left Outer Join, the data is not persisted. Instead, it is a view which has data as a result of the join on the multiple Persistent MDOs.

Before you create a MDO on the Workbench, ensure that the MDO Connector is enabled.

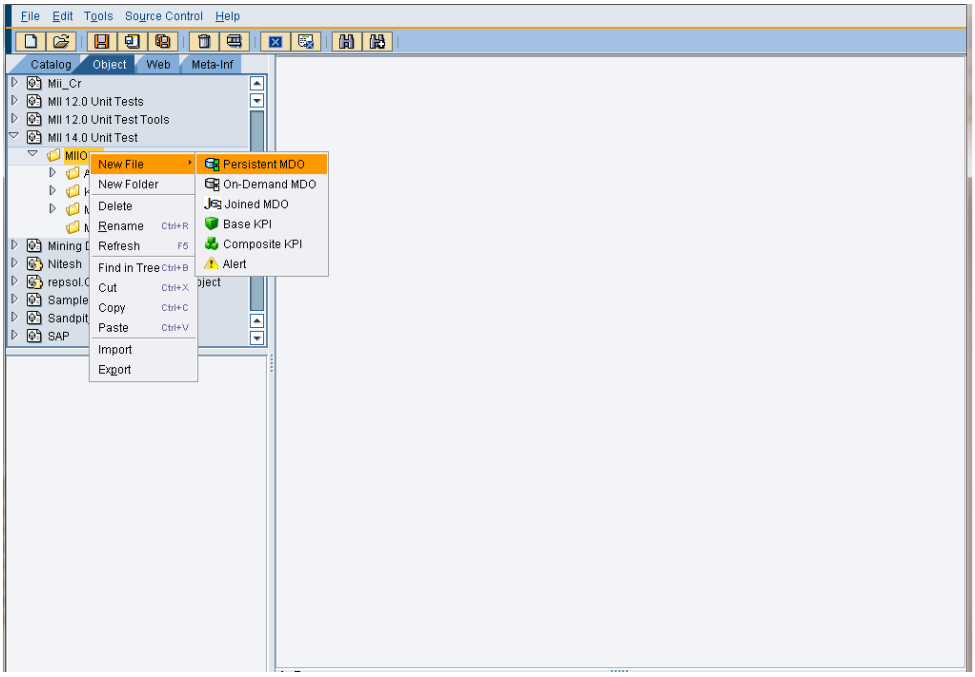
- Go to the SAP MII Administration screen -> Data Services -> Data Servers
- If MDO Connector not enabled, then select the *MDO Connector* and click on '*Enable'*.




## 1.1 Persistent MDO Definition

To create a new Persistent MDO in the SAP MII Workbench

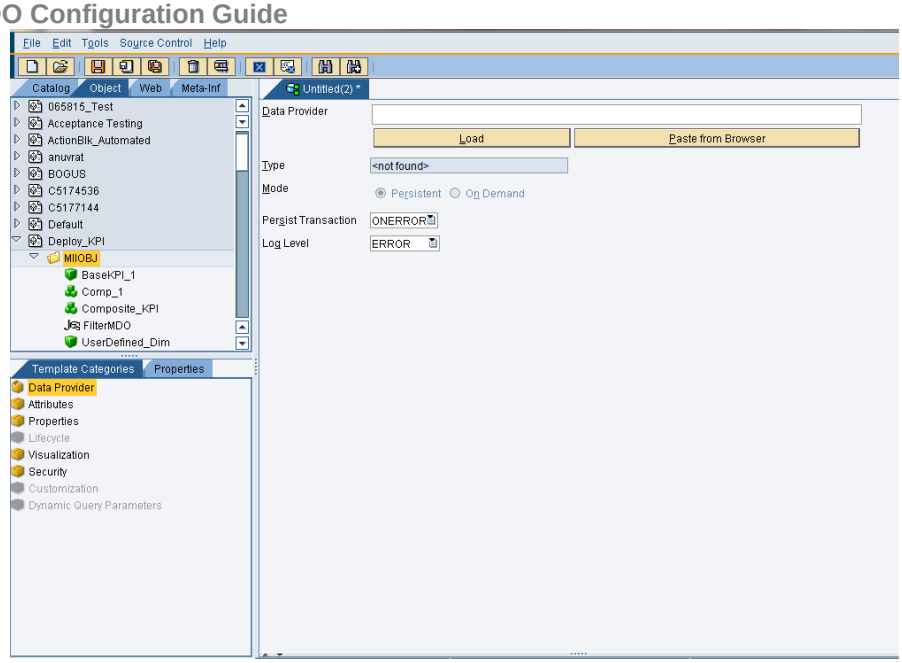
**Step 1**: Under the *MIIOBJ* folder of the *Object* tab, choose *File →New→Persistent MDO.*




### 1.1.1 Data Provider

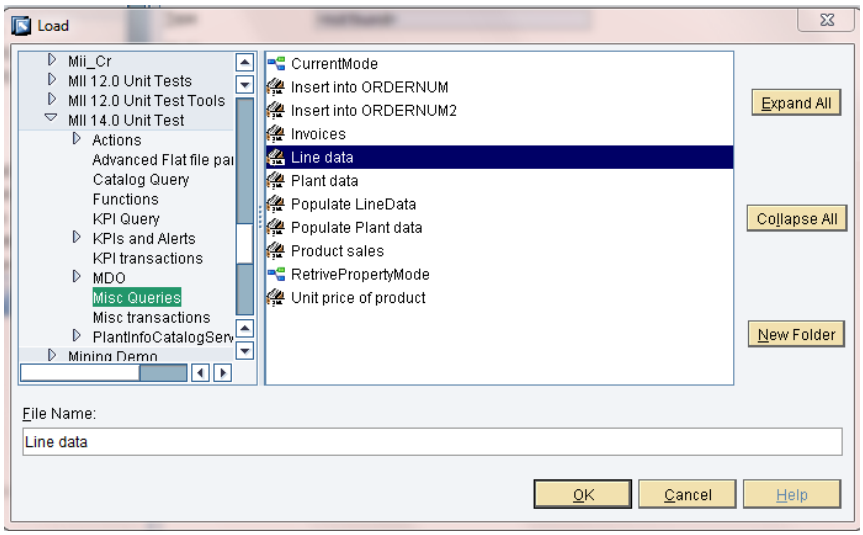
Data Provider is the data acquisition method used to define and return data for the object. You can only specify query templates or transactions as data providers for an MDO.

**Step 2**: On the *Template Categories* tab, go to the *Data Provider* screen, enter your **Transaction** or **Query Template** name for the MDO. The type of data provider appears in the *Type* field.

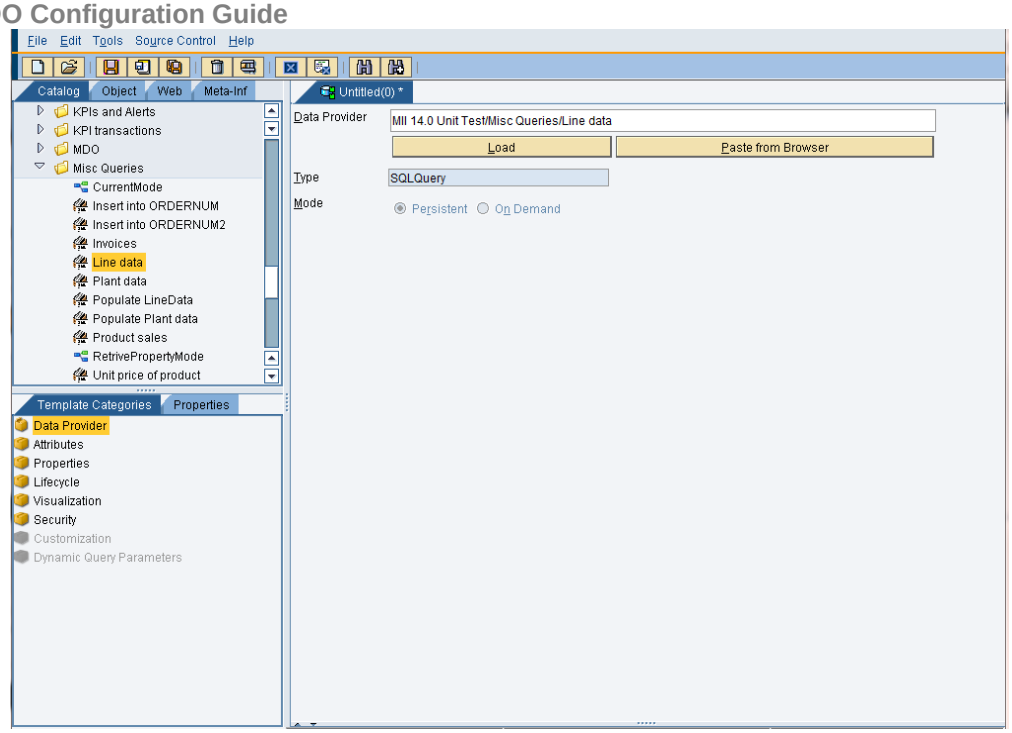



(i) Consider a SQL Query as the Data Provider for the Persistent MDO which can be loaded in two ways as follows:

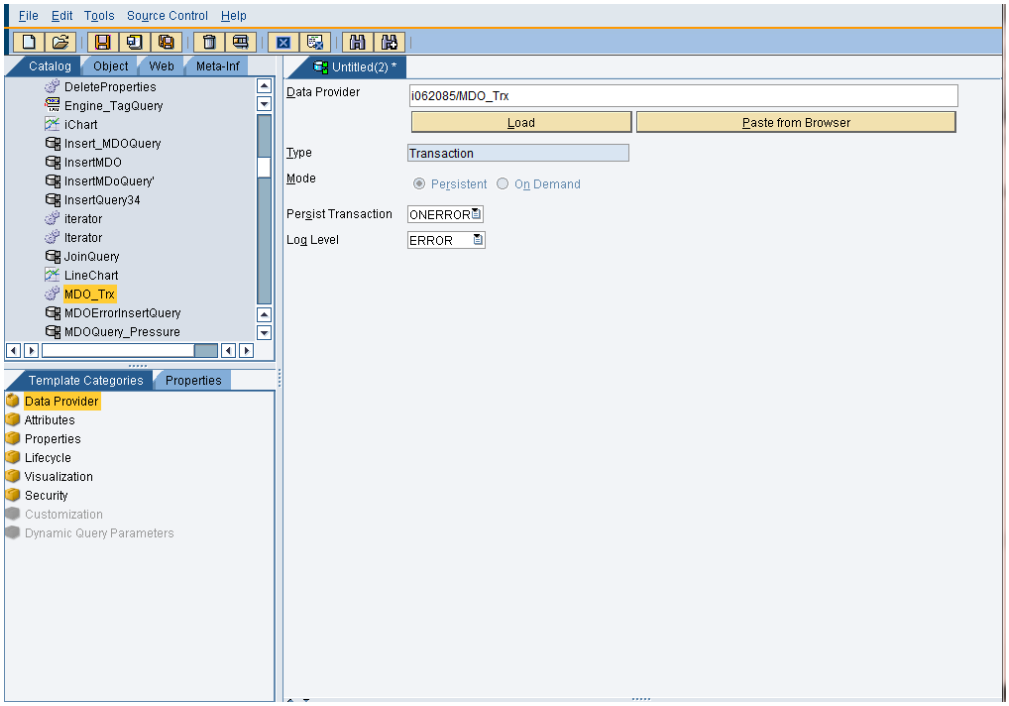
- Click on the "Load" button and select the Query Template from the Catalog File Browser .




- On click of "Paste from Browser", the currently selected Query Template that is highlighted in the *Catalog* File Browser will be pasted in the Data Provider Text Field.




(ii) Consider a Transaction as the Data Provider for the Persistent MDO.




The Persist Transaction property controls the persistence of the transaction into the Transaction Manager Database .Hence the user can select from the following options:

- OnError(Default) : Only on error it will persist the transaction
- Always : It will always persist the transaction
- Never : It will never persist the transaction

You may provide the *Log Level* ,which will make the logs entry based on the Log level that is set. By default it will set to' Error'.

The value can be one of the following:

- Unknown
- None
- Debug
- Info
- Warn
- Error
- Fatal

### 1.1.2 Attributes

**Step 3**: On the *Template Categories* tab, go to the *Attributes* screen to define the attributes (metadata) of the Data Provider.

**Step 4:** The '*Attributes'* screen will contain a column picker to provide the list of available attributes for the configured data source.
If a data source was configured, then the metadata for that data source will be retrieved.

(i) If the data source is a query template, then the query will be executed with to get the Column Attributes.
(ii) If a data source is a transaction, then the Transaction is executed get the list of outputs for the transaction.

The user can add Attributes by using the buttons provided '<','<<'to add from the right column names of the Data Provider to the left .
You can use the *Source Mapping Type* radio buttons to specify whether you want to map the attributes by name or position.

*Name mapping*
Maps the MDO attribute to the data provider attribute by name. This source name of the attribute must match the data provider attribute name.

*Position mapping*

Maps the MDO attribute to the data provider attribute by the position of the attribute in the dataset returned by the data provider.

In order to create Custom Attributes , you can create via New Button and observe that the in the details of the '*Selected Attribute'* the check box near the Custom Attribute is selected and the *Source Name* and *Source Position* are disabled.

*Details of the Selected Attribute*

You can modify the attribute details.

**Name** : Name of the Attribute

**Description**: Description of the Attribute

**Type:** Defines the MDO attribute type as one of the following:

- Boolean
- Byte
- dateTime
- double
- float
- integer
- long
- short
- string

**Length:** Defines the length of the MDO Attribute

**Default Value:** Defines a default value for an MDO attribute. If a value is not provided by the data provider, this default value is used.

**Min and Max Value:** Specifies the Minimum and Maximum values of the MDO attribute.

**Key:** Specifies whether the MDO Attribute is the unique identifier for the MDO. You can specify multiple attributes as keys.

**Custom Properties:** Additional metadata for an MDO attribute. Each property is defined by a key and an associated value. The properties are returned and visible when the object XML is generated.

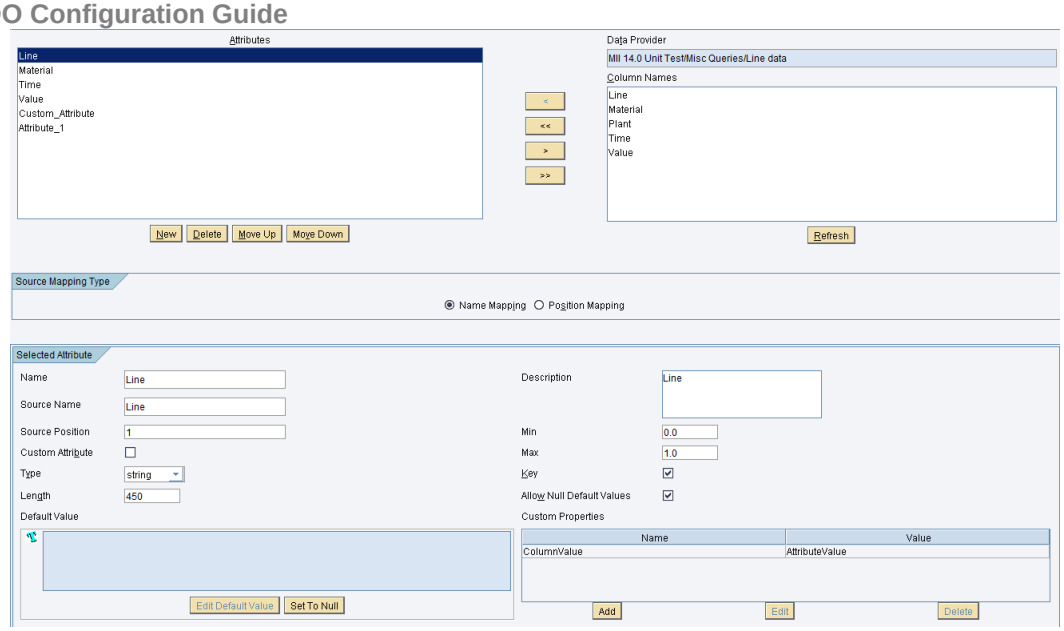
**Example:**

- ➔ Select the Attributes '*Line', 'Material', 'Time'* and *'Value'.*
- ➔ Create a Custom Attribute '*Custom_Attribute* ' and provide a default value to it by clicking on the **Edit Default Value'** button which displays a pop up where the user can enter the default value.

The '**Set to Null'** Button sets the default value for the attribute to null.

**Note*: The 'Edit Default Value' button will be enabled for Custom Attributes Only. The 'Set to Null' button will be enabled for all the Attributes in order to set the existing default value to null.**

- ➔ Select Attribute '*Line'* and select the check box '**Key**' to set it as a primary key
- ➔ Select the checkbox '**Allow Null Default Values'** where when checked the attribute can have null values
- ➔ Add a Custom Property lets say with a name 'CustomValue' and value 'AttributeValue'.




### 1.1.3 Properties

On the *Template Categories* tab, go to the *Properties* screen to define the following for your MDO :

- *Description*

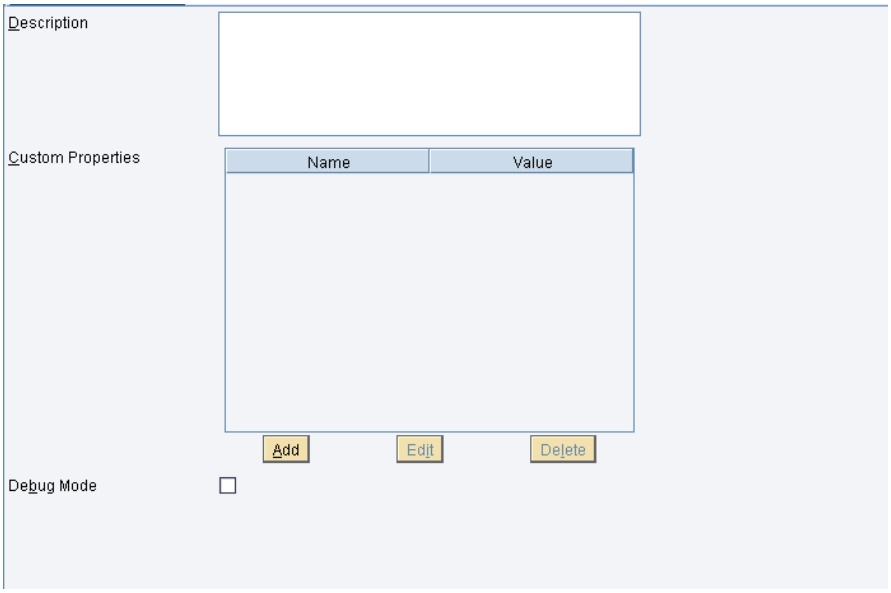
  You can use this field to define your Persistent MDO.

- *Custom Properties*

  You can define additional metadata for your MDO. Each property is defined by a key and associated value. These properties are returned and visible when the object XML is generated.

- Debug Mode

  You can use this field to enable the additional logging information for each MDO.



### 1.1.4 Lifecycle

The lifecycle is used for creating a scheduled job on the MDO to retrieve data from the data providers at scheduled time pattern specified and persist it in the MDO. You can perform various operations by running the scheduled job on the MDO like updating the existing data or replacing it entirely or deleting the persisted data. Even transactions can be executed after the scheduled job is run on the MDO.

**Step 5:**

**(i)** On the *Template Categories* tab, go to the *Lifecycle* screen to define tasks to update, replace or delete the persisted data.

You can configure it with the following properties:

**Name** : Name of the task

**Description**: Description of the task

**Enabled**: Checkbox that indicates whether a task is enabled for execution or not.

**Credential Alias**: To authorize the execution of a task, you must specify a valid credential alias. The dropdown lists all the credential aliases which are populated on the Credential Store Screen. For more information about credentials, see Using Credential Stores..

**Pattern:** You can click the clock button to open the *Pattern Editor* screen where you define the schedule for running the task. You can schedule a task to run daily, hourly, weekly, monthly, and so on.

**Mode:** You can configure the task to do one of the following to the persisted data:

- ***Replace:*** When the MDO data provider is executed, the persisted data is deleted and replaced by the resulting data set.
- ***Update:*** When the MDO data provider is executed, the persisted data is updated with the resulting data set, which is based on the key attributes for the MDO.
- ***Delete:*** When the MDO data provider is executed , calls a Delete object query to clean up specified data.

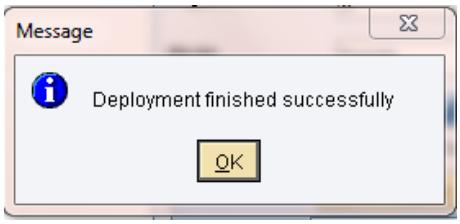
**Inputs:** The *Inputs* tab allows you to specify filter values for MDO attribute parameters, which are used when the task is executed. This function allows you to update, replace, or delete certain records in the persisted data.

**Postexecution:** When the task is complete, you can execute a transaction. You specify this transaction on the *Postexecution* tab. You can click the *Load* button to find the transaction. You can also enter values for input parameters for the transaction.

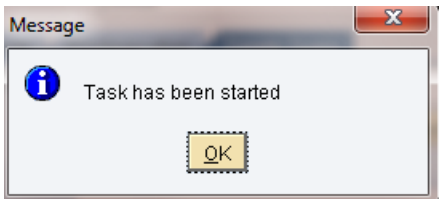
**Delete Query:** On the *Delete Query* tab, you can specify an existing MDO delete query to be used by your delete task.

To enable all tasks, you can click the *Enable All* button. To disable all tasks, you can click the *Disable All* button.
To execute the task, click on '*Deploy All'* and then click '*Run'*.

- On click of '*Deploy All',* a dialog box appears as shown below:



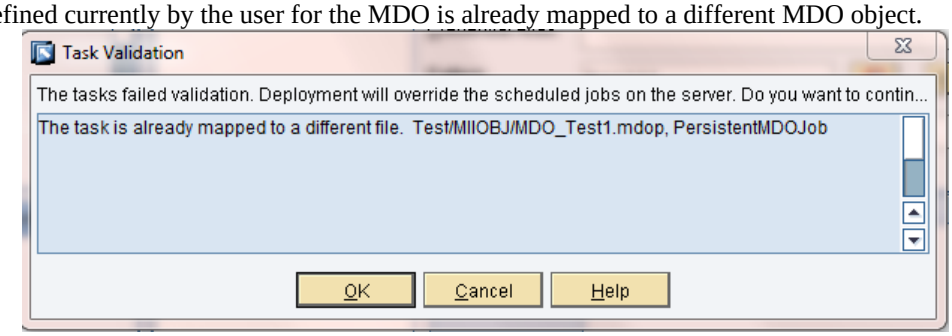

-On click of '*Run',* a dialog box appears as shown below:




***Existing Job with the same name***

Always ensure that the job / task defined for a MDO Object should be unique .And in order to avoid name clashes it is best advised to use a naming convention like the ***[MDOName}_[TaskName]*** (MDO Name followed by the task name to ensure uniqueness).

But nevertheless if a situation does occur then on Click of '*Deploy All',* a dialog box informing the user that the task defined currently by the user for the MDO is already mapped to a different MDO object.



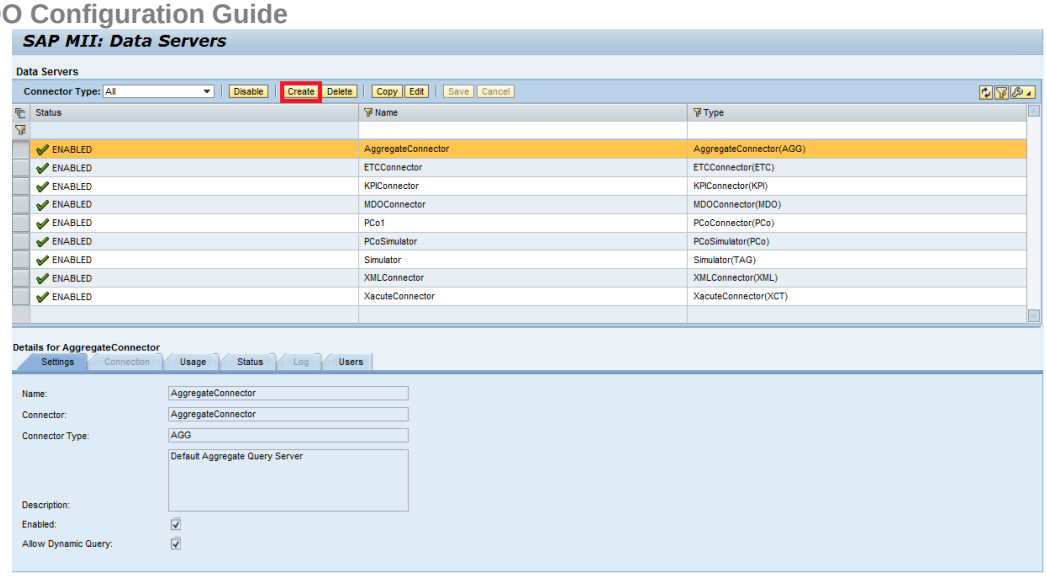

On click of 'Ok' the job with that name is mapped to this MDO defined by the user.
If not, then the user can click 'Cancel' to provide a different name to the task/job.

The entry will be made in ***Scheduler*** screen of the **SAP MII Administration** screen where the user can later update the task.

**(ii)** You can also create a Scheduled Job from the SAP MII Administration.

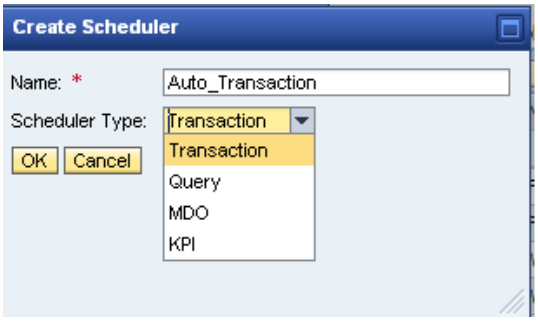
To create a scheduled job, do the following:

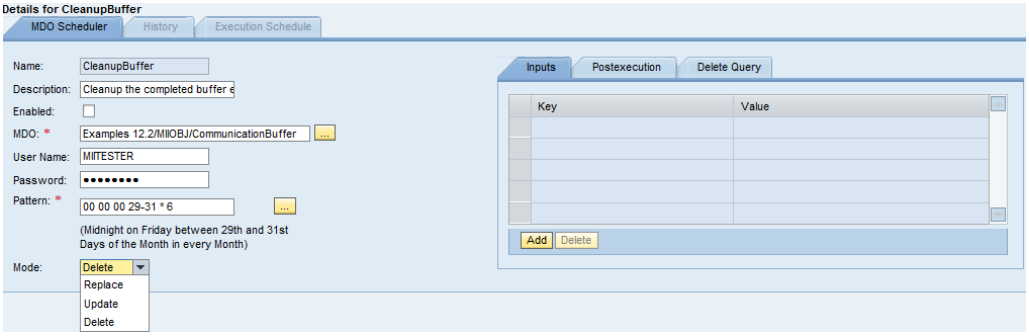
1. On the SAP MII administration menu, choose *System Management* →*Scheduler*.



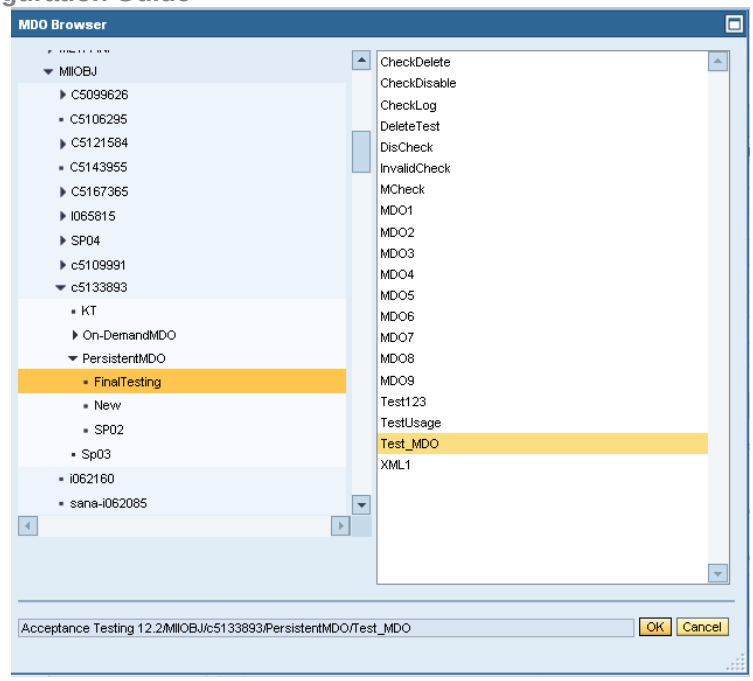

2. Click *Create*.

3. A Pop Up appears where you can give the 'Name' of the scheduler and the 'Scheduler Type' as '*MDO'* from the dropdown. Click on ok.



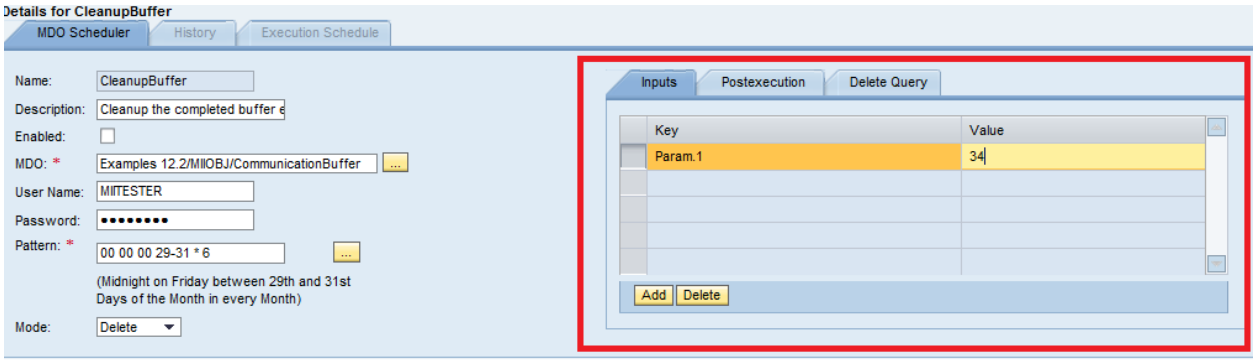

- 'Description'
- 'Enabled' check box may be checked if you would want the scheduler to be enabled and running at the scheduled time specified.
- Load any MDO Object by clicking on the browser button provided which displays a pop up where you may browse to the MDO objects contained in the workbench and select the MDO Object you want the scheduler to execute. Click on Ok .



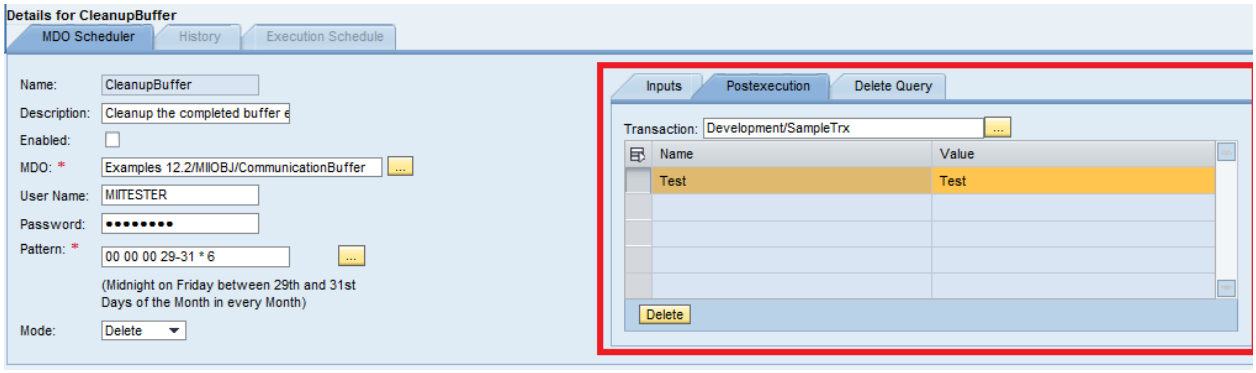

- Provide the ‘User Name’ and ‘Password’
- ‘Pattern’ where on click of the load button a pop up appears and provide the pattern at which the scheduler will run .Click on Ok.
- ‘Mode’ to specify the type of mode to be performed namely ‘Replace’ ,’Update’ and ‘Delete’
    - ‘Replace’ mode will replace all the data in the result set of the MDO object with the new data
    - ‘Update’ mode will update the current result set of the MDO object
    - ‘Delete’ mode will allow the user to lad the delete MDO query to perform operations of deletion on the data of the MDO object

- The Table just adjacent in the ‘Details’ Tab allows the user to perform the following :

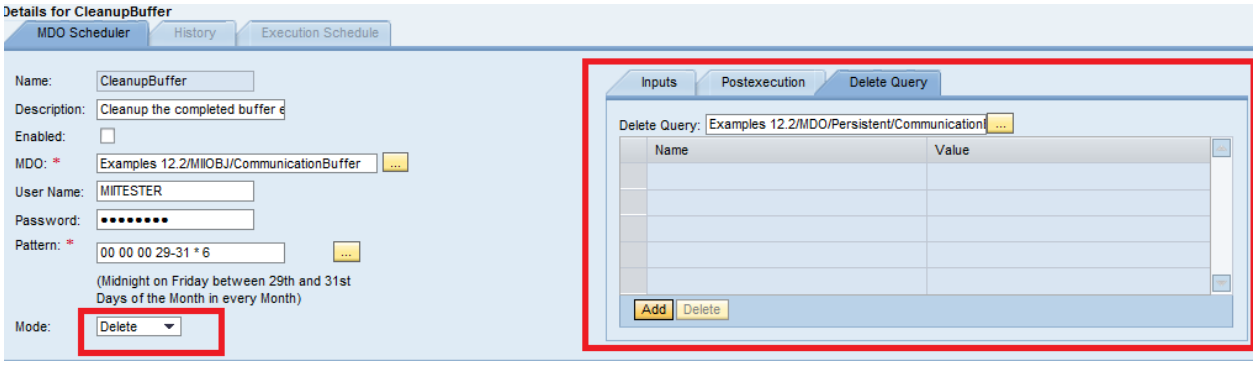
o Provide input parameters by clicking on ‘Add’ and give the Parameter key and the value.




o In the ‘Postexecution’ tab, where you can load a transaction by the browse button provided and the parameters of the transaction will be displayed in the table below it (‘Name’ and ‘Value’ pairs).




- The 'Delete' tab is enabled only when the mode 'Delete ' is selected. Load a MDO Query template of Delete mode. Click on 'Add' to add the parameters to provide the input paramter for the Delete MDOQuery.

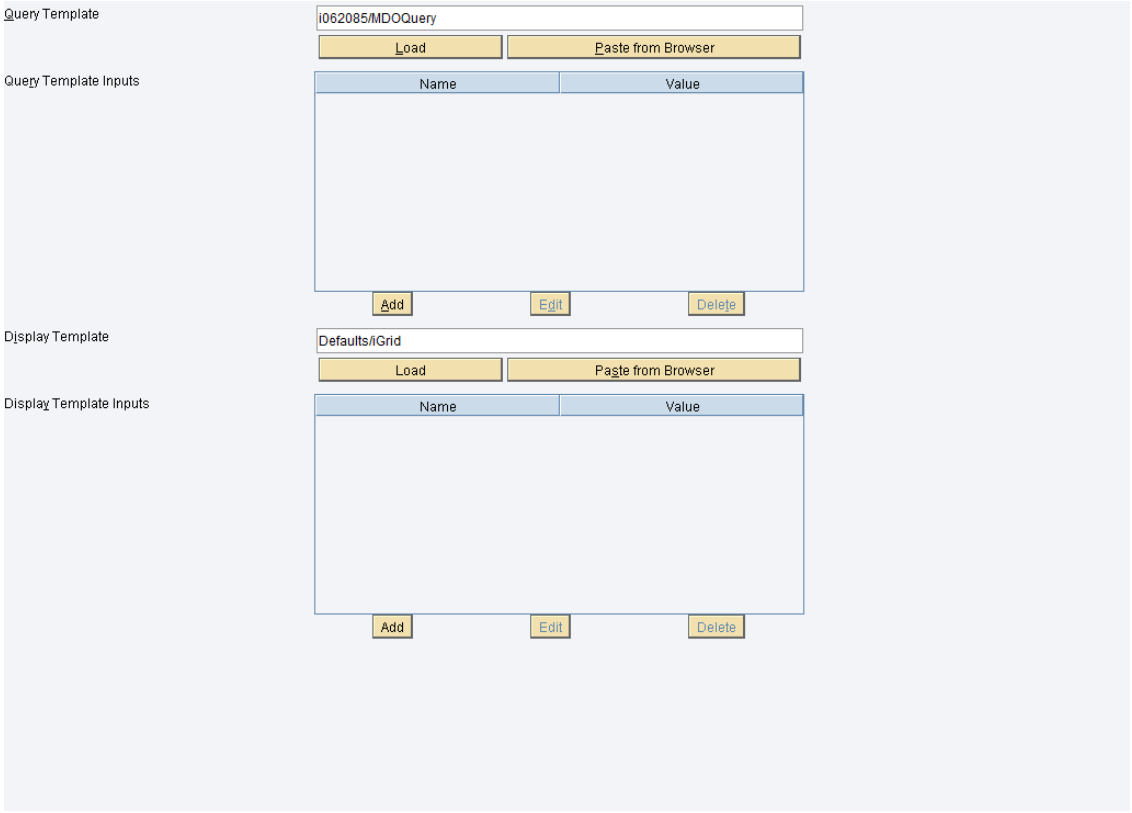



4. Enter the required data and click *Save*.
5. Now click on 'Run' to run the scheduler.

### 1.1.5 Visualization

The *Visualization* screen, is used to specify the display of your MDO, when it is executed. You can specify a MDO Query Template and Display Template to present the results of the MDO.

This visualization is used to display the results in form of charts and hence can be used in reporting.

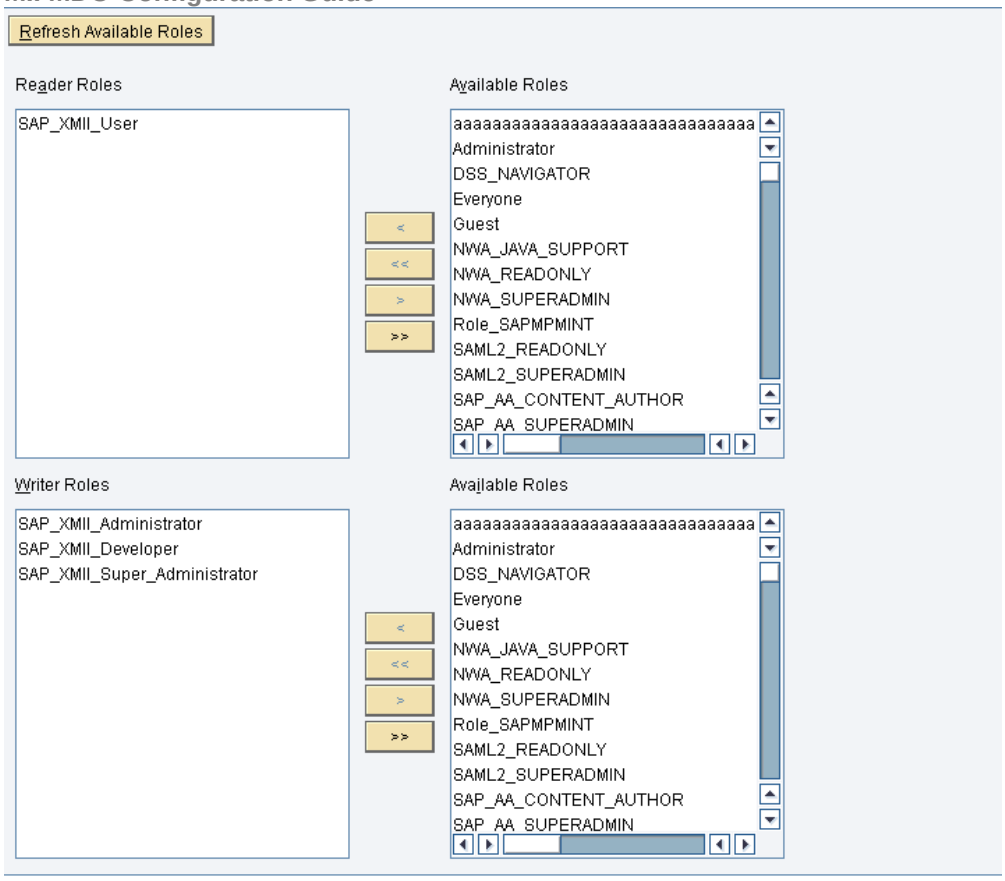



### 1.1.6 Security

You use this screen to authorize roles to read or write to a MDO.

You can select the roles from the '*Available Roles'* column and add it to the '*Reader Roles'* and '*Writer Roles'* by clicking on '<' or'<<' to select all.

Hence only the roles specified in the '*Reader Roles'* and '*Writer Roles'* will have the respective authorizations to read or write respectively to a MDO.

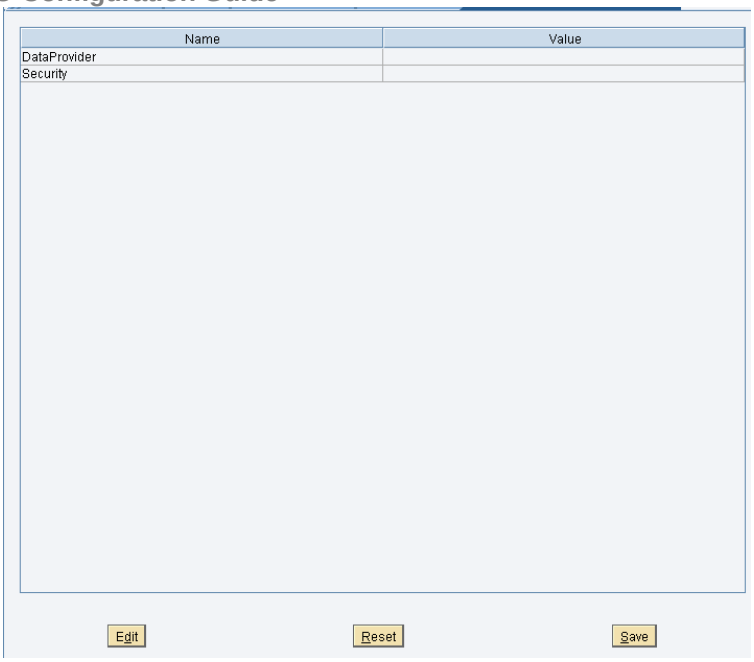



### 1.1.7 Customization

On the *Customization* screen, you can customize an MDO that is part of a project deployed by NWDI source control. If the MDO is a component of a deployed project, you can modify the configuration of the data provider and security roles.

The MDO in a deployed Project is Read-Only and none of the panels in the *Template Categories* tab are enabled except the *Customization* tab.

Go to the *Customization Tab* and you can see there are two entries namely: Data Provider and Security

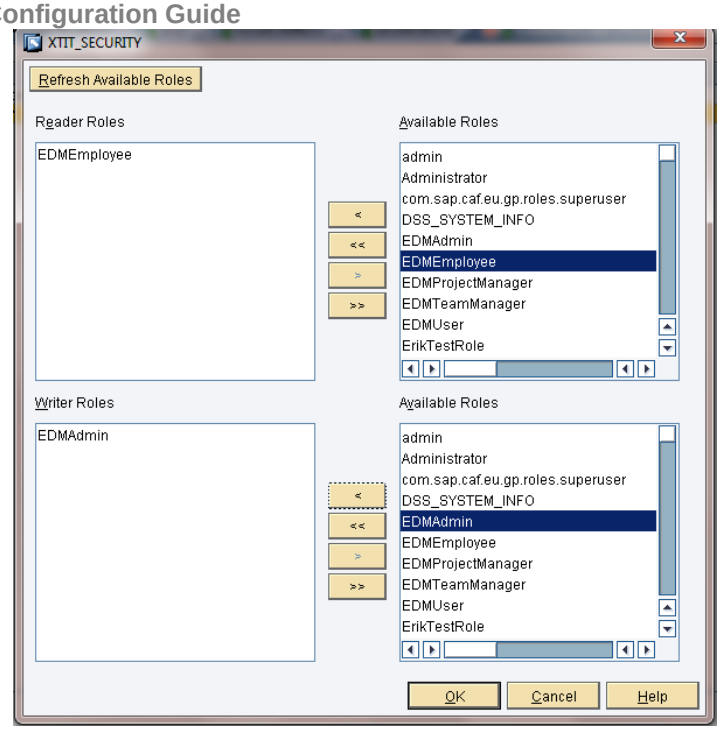



***Data Provider Configuration***

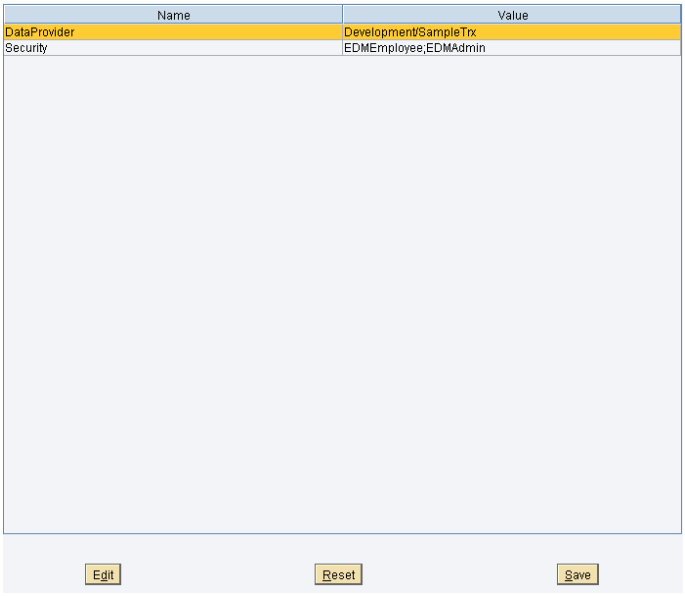
- Select' *Data Provider'* in case if you would want to change the data provider for the MDO and Click on '*Edit'*.
- A dialog box appears ,where the user can load the Query Template/Transaction from the File Browser.

***Security Configuration***

- Select '*Security*' in case if you would want to change the reader/ writer roles to the MDO and Click on '*Edit'*.
- A dialog box appears , where the user can select the reader writer roles.




- Click on ok and the Roles defined or /and Query Template /Transaction defined will appear against the Security and Data Provider fields respectively.

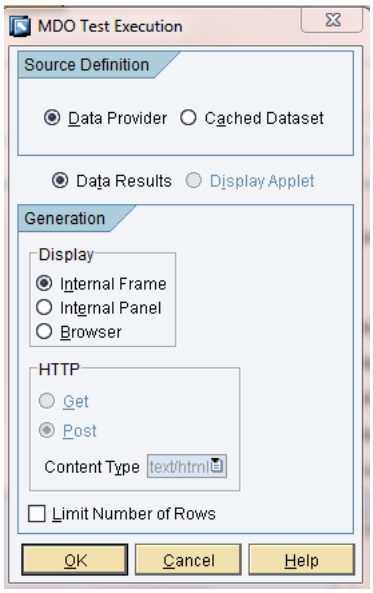



- Click on save to save the update the customizations to the MDO/ reset if you would want to revert the values .

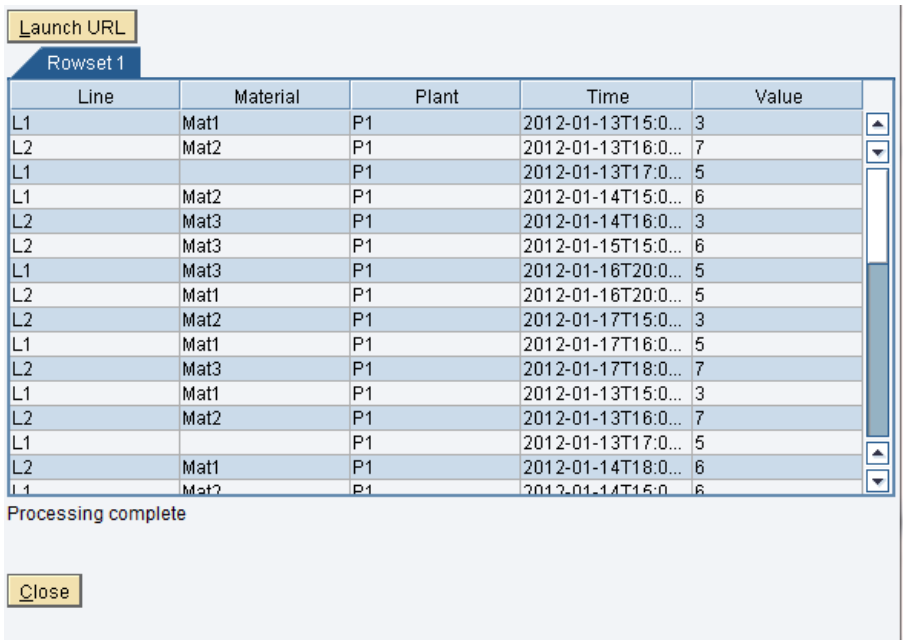
**Step 6** : Save the *Persistent MDO* by clicking on *File-> Save.*

**Step 7**: To test the Persistent MDO, choose *Test.*

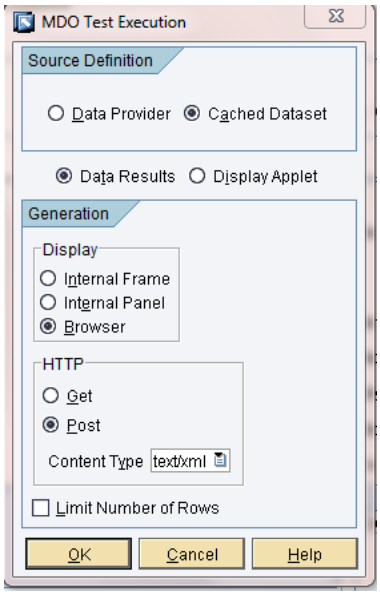
(i) A dialog is displayed, with a radio button that allows the user to use the execution of the Data Provider. You can test the MDO by executing and displaying the data in the Internal frame ,Internal Panel or by launching the browser displaying thit in the html, xml or csv format format as shown below.



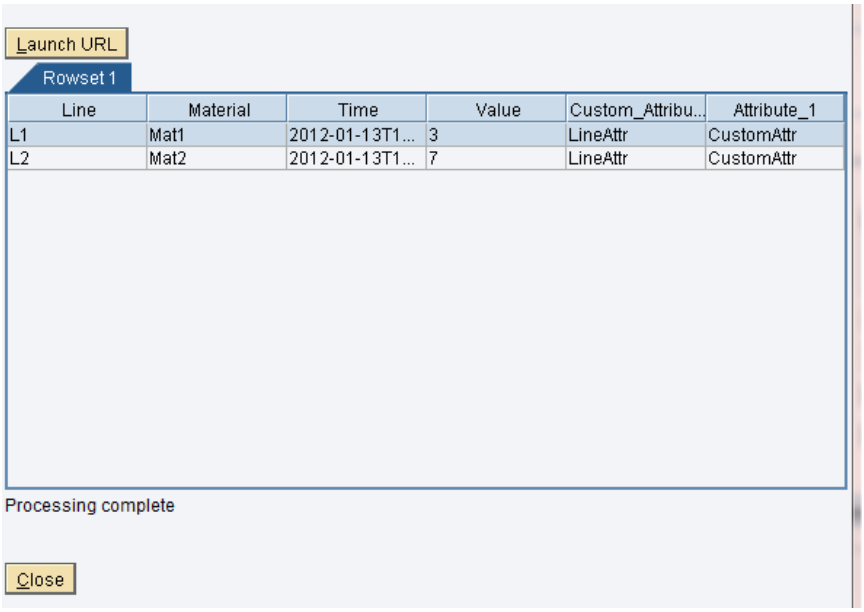

On Click of Ok , the following result appears



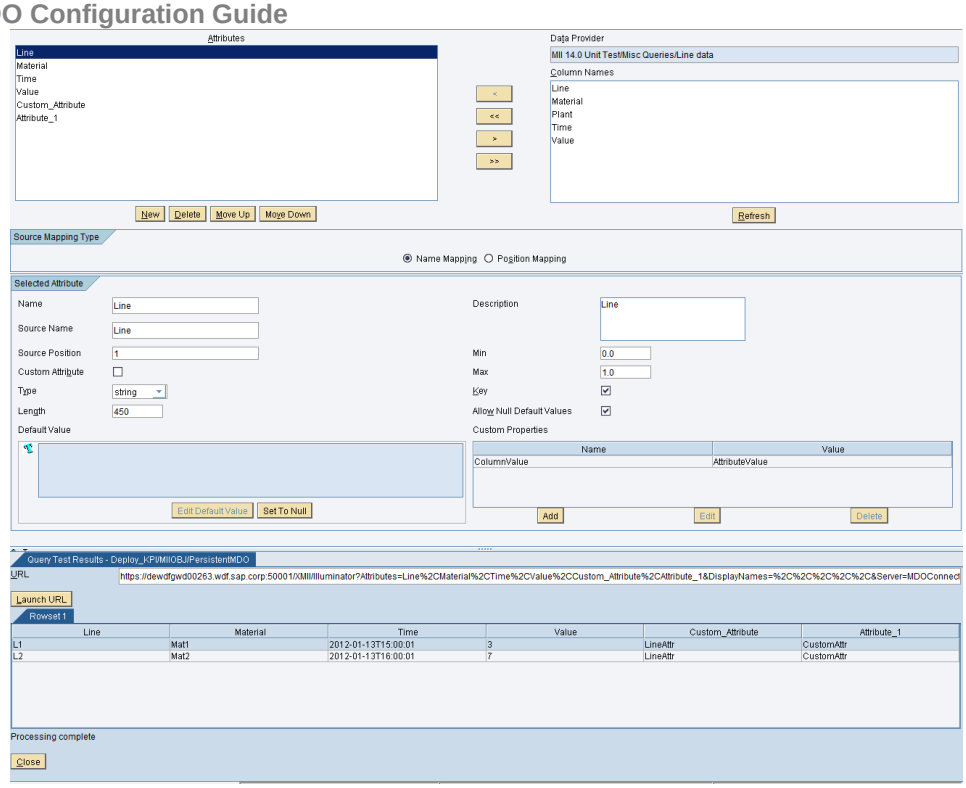

(ii) With a radio button that allows the user to use the execution of the C*ached Dataset.* You can test the MDO by executing and displaying the data in the Internal frame ,Internal Panel or by launching the browser displaying thit in the html, xml or csv format format as shown below.




On execution of the MDO in the Internal Frame




On execution of the MDO in the Internal Panel




On execution of the MDO by launching the browser with content type text/xml

In the result shown below(browser) , you can observe that the Custom Attributes defined by the user namely: Name :ColumnValue and Value:AttributeValue is defined in the xml generated.

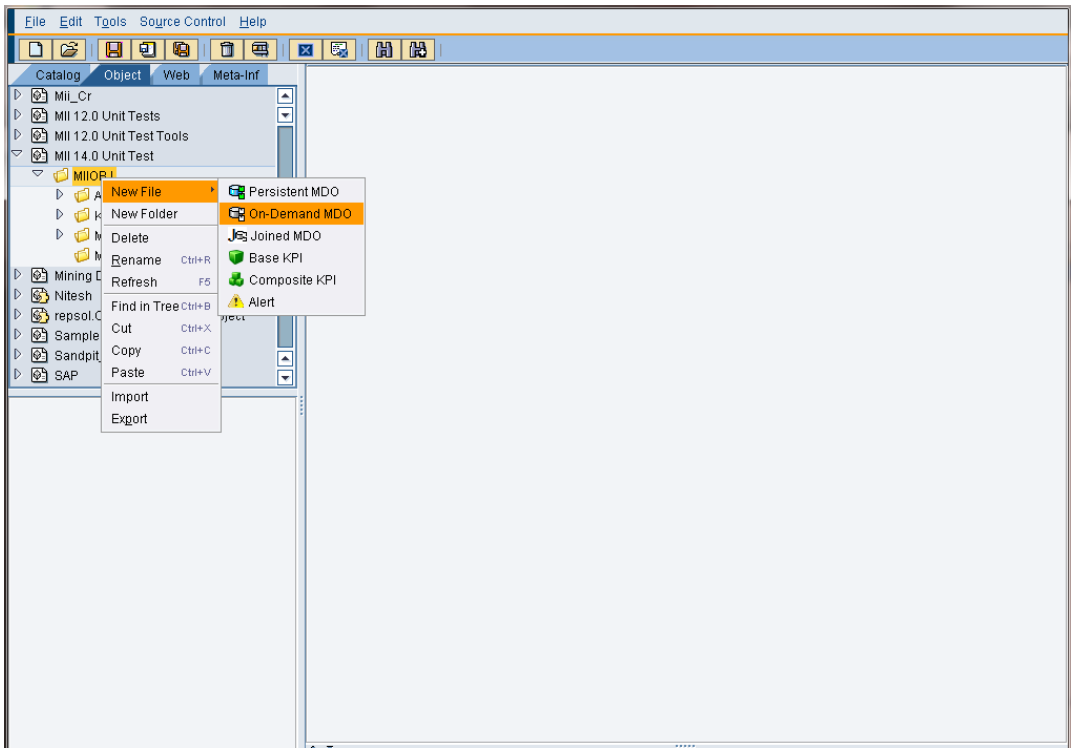
```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <Rowsets CachedTime="" DateCreated="2012-10-10T12:44:55" EndDate="2012-10-10T12:44:55" StartDate="2012-10-10T11:44:55" Version="14.0.0 Build(780)">
  - <Rowset>
    - <Columns>
        <Column ColumnValue="AttributeValue" Description="Line" MaxRange="1" MinRange="0" Name="Line" SQLDataType="12" SourceColumn="Line" />
        <Column Description="Material" MaxRange="1" MinRange="0" Name="Material" SQLDataType="12" SourceColumn="Material" />
        <Column ColumnAttr="AttributeValue" Description="Time" MaxRange="1" MinRange="0" Name="Time" SQLDataType="93" SourceColumn="Time" />
        <Column Description="Value" MaxRange="1" MinRange="0" Name="Value" SQLDataType="8" SourceColumn="Value" />
        <Column Description="" MaxRange="100" MinRange="0" Name="Custom_Attribute" SQLDataType="12" SourceColumn="Custom_Attribute" />
        <Column Description="" MaxRange="100" MinRange="0" Name="Attribute_1" SQLDataType="12" SourceColumn="Attribute_1" />
      </Columns>
    - <Row>
        <Line>L1</Line>
        <Material>Mat1</Material>
        <Time>2012-01-13T15:00:01</Time>
        <Value>3</Value>
        <Custom_Attribute>LineAttr</Custom_Attribute>
        <Attribute_1>CustomAttr</Attribute_1>
      </Row>
    - <Row>
        <Line>L2</Line>
        <Material>Mat2</Material>
        <Time>2012-01-13T16:00:01</Time>
        <Value>7</Value>
        <Custom_Attribute>LineAttr</Custom_Attribute>
        <Attribute_1>CustomAttr</Attribute_1>
      </Row>
    </Rowset>
</Rowsets>
```

(iii) You can also view the results with the Display template you specified in the *Visualization* screen by selecting the '*Display Applet'*.

## 1.2 On-Demand MDO Definition

To create a new On-Demand MDO in the SAP MII Workbench

**Step 1**: Under the *MIIOBJ* folder of the *Object* tab, choose *File →New→On-Demand MDO.*




### 1.2.1 Data Provider

This screen is same as the existing Data Provider tab of Persistent MDO object. Refer Persistent MDO Data Provider

### 1.2.2 Attributes

This screen is same as the existing visualization tab of Persistent MDO object. Refer Persistent MDO Attributes

The Lifecycle Tab is disabled for On-Demand MDOs since the data is obtained on real –time and hence not persisted.

### 1.2.3 Visualization

This screen is same as the existing Visualization tab of Persistent MDO object. Refer Persistent MDO Visualization

### 1.2.4 Security

This screen is same as the existing Security tab of Persistent MDO object. Refer Persistent MDO Security

### 1.2.5 Customization

This screen is same as the existing Customization tab of Persistent MDO object. Refer Persistent MDO Customization

### 1.2.6 Dynamic Query Parameters

On this screen ,the user can modify the Data Provider execution at runtime where the parameters are passed from the MDO Query to On-demand MDO .For more information refer Object Parameters

**Step 4** : Save the *On-Demand MDO* by clicking on *File-> Save.*

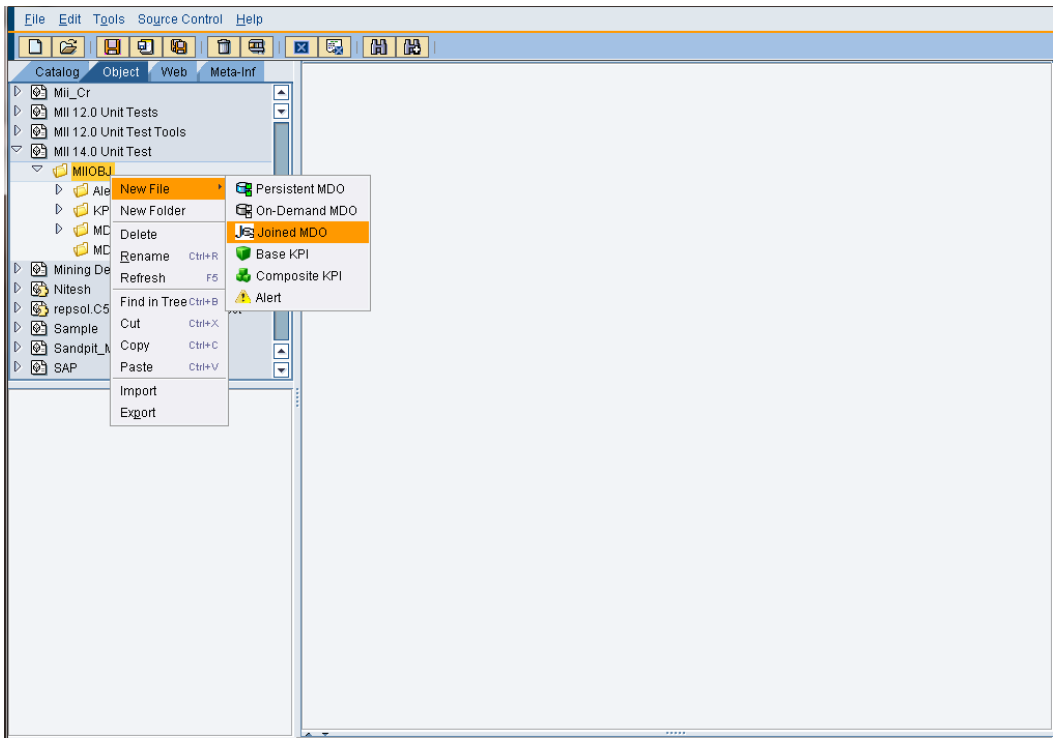
**Step 5**: To test the On-Demand MDO, choose *Test.*

Result similar to Step 7 of Persistent MDOs.

## 1.3 Joined MDO Definition

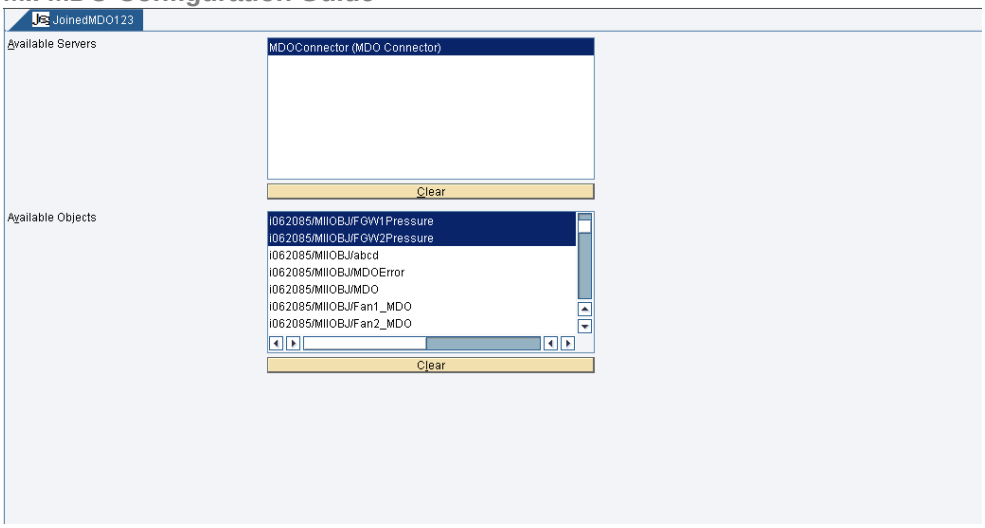
To create a new Joined MDO in the SAP MII Workbench

**Step 1**: Under the *MIIOBJ* folder of the *Object* tab, choose *File →New→Joined MDO.*




### 1.3.1 Data Source Panel

**Step 2**: On the *Template Categories* tab, go to *Data Source screen* and select the MDO Connector available in the '*Available Servers'* list and select two or more Persistent MDOs.




### 1.3.2 Attributes Panel

**Step 3**: Go to the *Attributes* screen and perform the following.

(i) Select a MDO from the '*Selected Objects'* list which populates the MDOs selected on the Data Source Panel and below it displays the attributes of the MDO

(ii) Select the attributes from the '*Available Attributes'* list .The selected attribute appears in the TextField below it. Click on 'Add'/drag to add the attribute or 'Add All' to add all the attributes in the '*Available Attributes'* list to the '*Selected Attributes'* table.

Example : Select the MDO *i062085/MIIOBJ/FGW1Pressure* from '*Selected Objects' and* select the attributes 'DateTime' and 'FGW1AccumFlow' from the '*Available Attributes'* list .Click on 'Add' button to add the selected attributes in the '*Selected Attributes'* table. Then select the second MDO *i062085/MIIOBJ/FGW2Pressure* from '*Selected Objects' and* select the attribute 'FGW2Pressure' from the '*Available Attributes'* list .Click on 'Add' button to add the selected attributes in the '*Selected Attributes'* table.

Note*: The '*Selected Attributes'* table consists of two columns namely:

*Selected Attributes* *:* Contains which attributes to select as the elements for each row. It contains the entire path location of the MDO and its attribute. This is non-editable field. Here in this example the selected Attributes are the entire paths of the respective attributes like *i062085/MIIOBJ/FGW1Pressure/DateTime, i062085/MIIOBJ/FGW1Pressure/FGW1AccumFlow and i062085/MIIOBJ/FGW2Pressure/FGW2Pressure.*

*Display Name:* It has the alias name of the selected attribute(editable field).When the query is executed , the aliases for the selected attributes will be displayed as the columns. Here in this example the display names are *DateTime , FGW1AccumFlow and FGW2Pressure.*

(iii) You can change the Display Names of the selected Attributes by double clicking on the table row of the Display Name and it becomes editable.

(iv) To add the Join Expression ,Click on 'Add' button / usage of the Secondary mouse click and select 'Add' from the context menu.

(v) Select an MDO from the first column that represents the first table say i0*62085/MIIOBJ/FGW1Pressure* . Select the drop down of join type and select a join(Inner Join and Left Outer Join) say for this example *INNER JOIN*. Similarly select the MDO from the dropdown of the third table column say *i062085/MIIOBJ/FGW2Pressure*

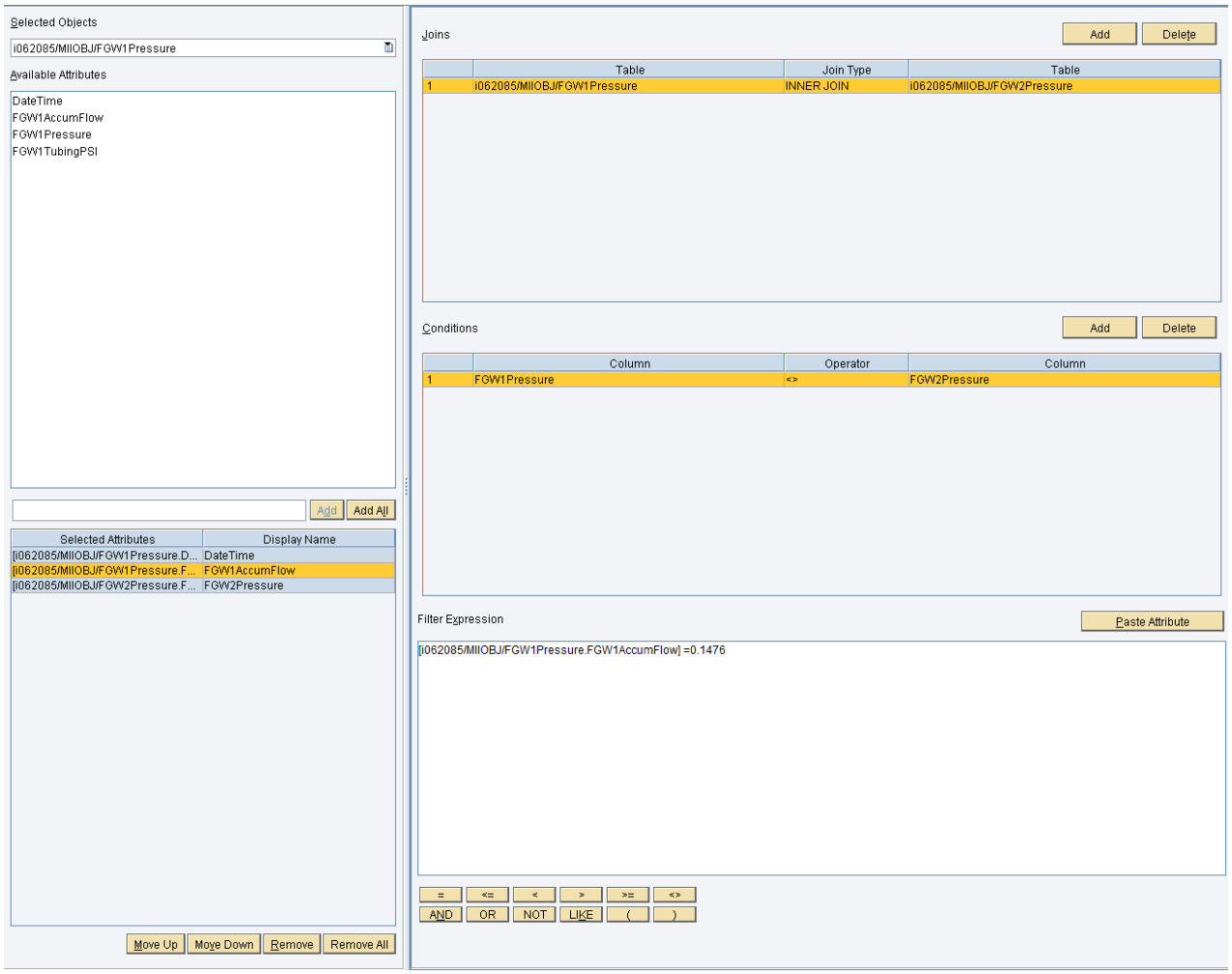
(vi) To add a condition to the respective Join. Select the join(in the '*Joins'* Table).Click on the 'Add' button /Secondary mouse click and select 'Add' from the context menu in the 'C*onditions'* Table and it adds a row in the respective table.

(vii) Select the first column which contains all the attributes of the MDO selected in the first column of the *Joins* table say *FGW1Pressure*. Select the operator say '<>' and an attribute in the third column from the list which contains the attributes of the MDO selected in the third column of the *Joins* table say *FGW2Pressure*.

(viii) Provide the Filter expression by dragging the attributes /selecting the attribute from the '*Available Attributes'* or the '*Selected Attributes'* table and clicking on '*Paste Attribute'* button.

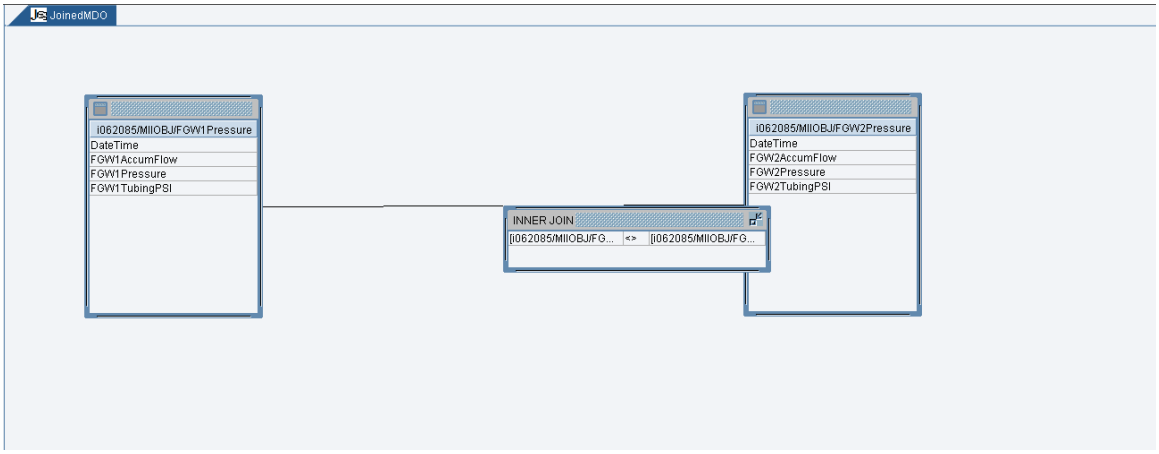
Hence say you select the second attribute from the '*Selected Attributes'* table namely *FGW1AccumFlow* and click on '*Paste Attribute'* button provided with the Filter area and give the expression followed by the pasted attribute as '=0.1476' hence the filter expression is

'*i062085/MIIOBJ/FGW1Pressure/FGW1AccumFlow = 0.1476'*




### 1.3.3 Layout Panel

The *Layout* screen displays a graphic view (ER Diagram) of Joins between selected MDO Objects. Any changes made on '*Attributes*' screen will be reflected on this Layout screen.




### 1.3.4 Visualization Panel

This screen is same as the existing visualization tab of Persistent MDO object. Refer Persistent MDO Visualization

### 1.3.5 Security Panel

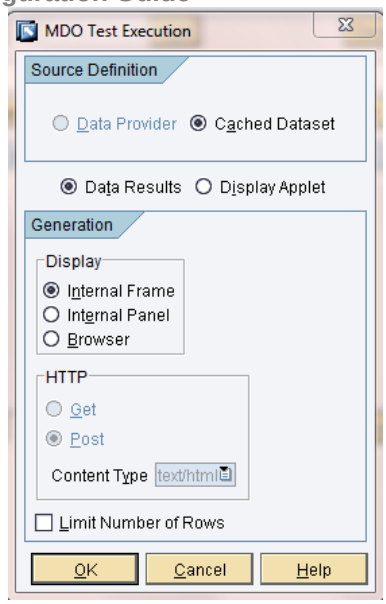
This screen is same as the existing visualization tab of Persistent MDO object. Refer Persistent MDO Security

**Step 4** : Save the *Joined MDO* by clicking on *File-> Save.*

**Step 5**: To test the Joined MDO, choose *Test*.

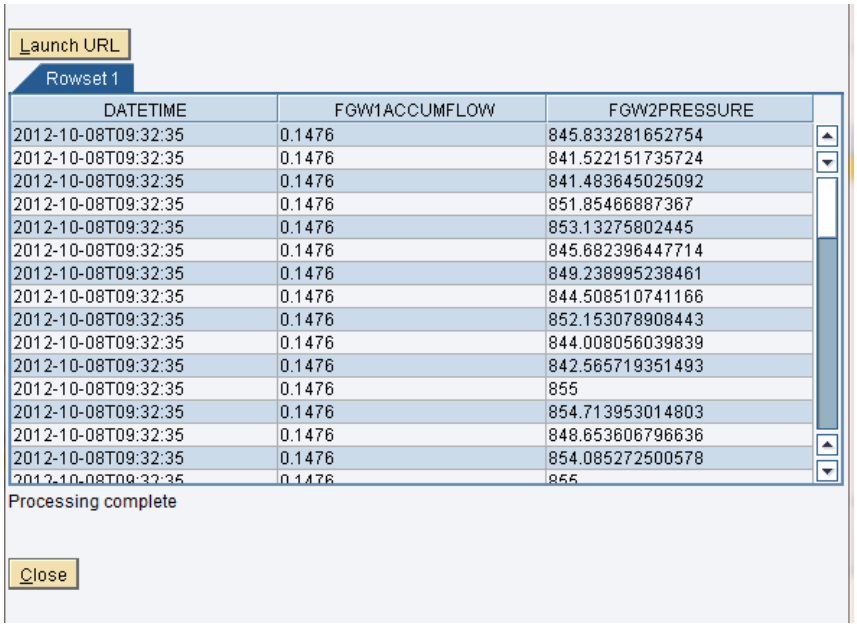
A dialog is displayed, with a radio button that allows the user to use the execution of the *Cached Dataset*

(iv) You can test the MDO by executing and displaying the data of the Joined MDO by selecting '*Data Results'* Option.

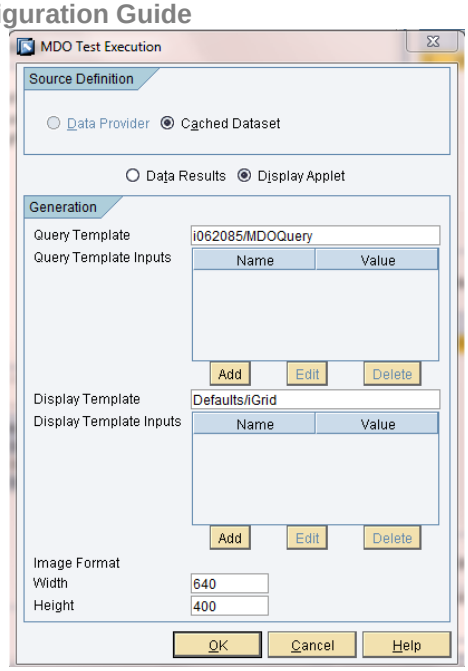



***Result:***

It returns the result which retrieves the joins performed on 'FGW1Pressure' MDO and 'FGW2Pressure' MDO of the Oil Tanks whose Accumulated Flow Rate of Valve 1 is"0.1476".

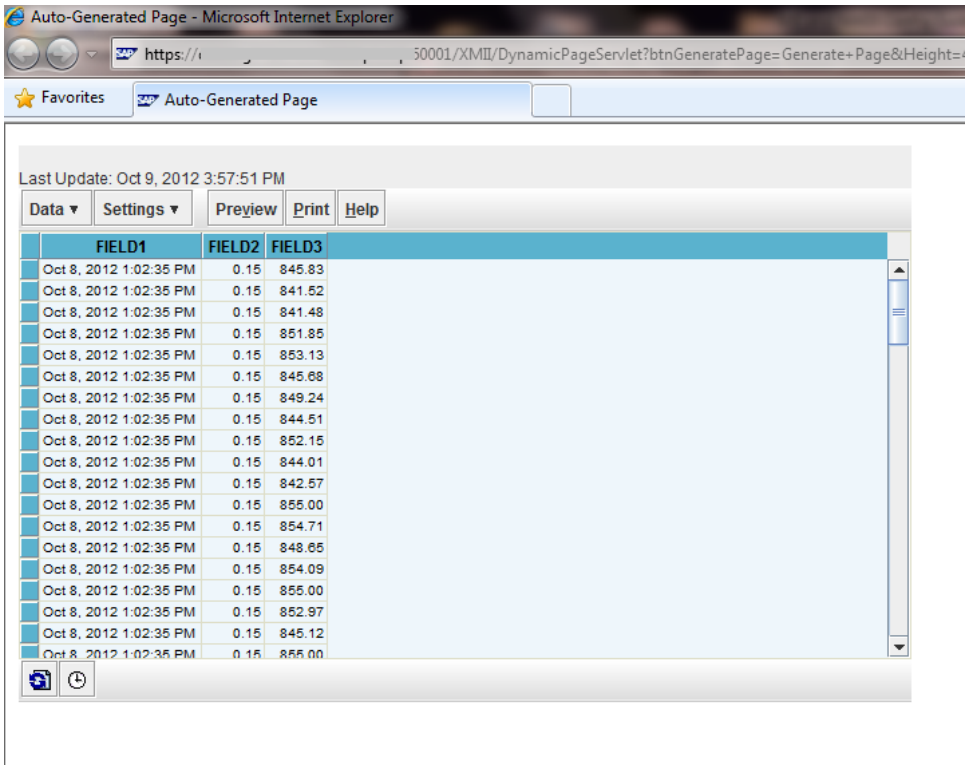



(v) You can also view the results with the Display template you specified in the *Visualization* screen by selecting the '*Display Applet'*.




***Result:***

It returns the result which retrieves the joins performed on 'FGW1Pressure' MDO and 'FGW2Pressure' MDO of the Oil Tanks whose Accumulated Flow Rate of Valve 1 is"0.1476" in the grid format.



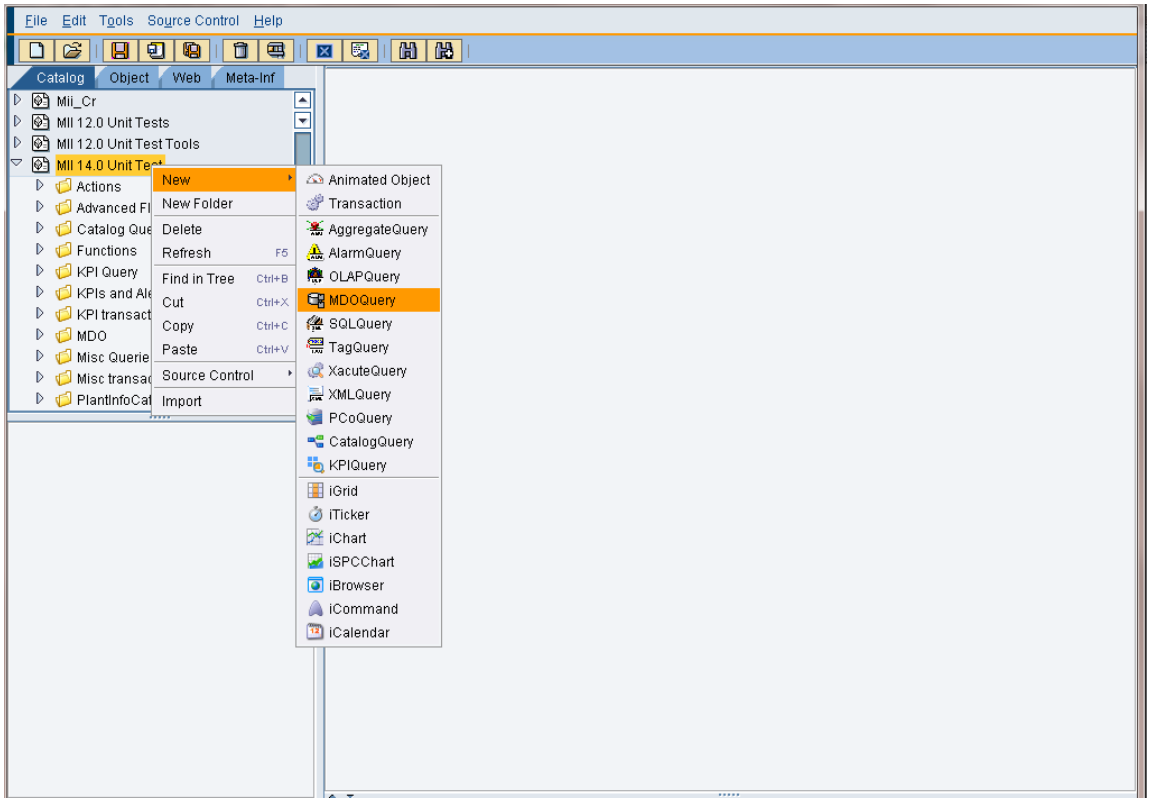
# 2. MDO Query Definition

This document describes the steps that will be required for the creation of the MDO Query. The types of Object queries will be as follows : SELECT,JOIN,INSERT,DELETE and UPDATE.

You use this function to do one of the following:

- Select and return the data associated with a manufacturing data object (MDO)
- Perform joins on two or more Persistent and/or Joined MDOs and return the data
- Insert, update, or delete the data persisted for the custom attributes of a persistent MDO

**Step 1:** Open the workbench, under the 'Catalog' tab, select a project-> Right-mouse click-> New-> MDO Query




**Step 2:** Select the 'Data Source' in the template categories provided.

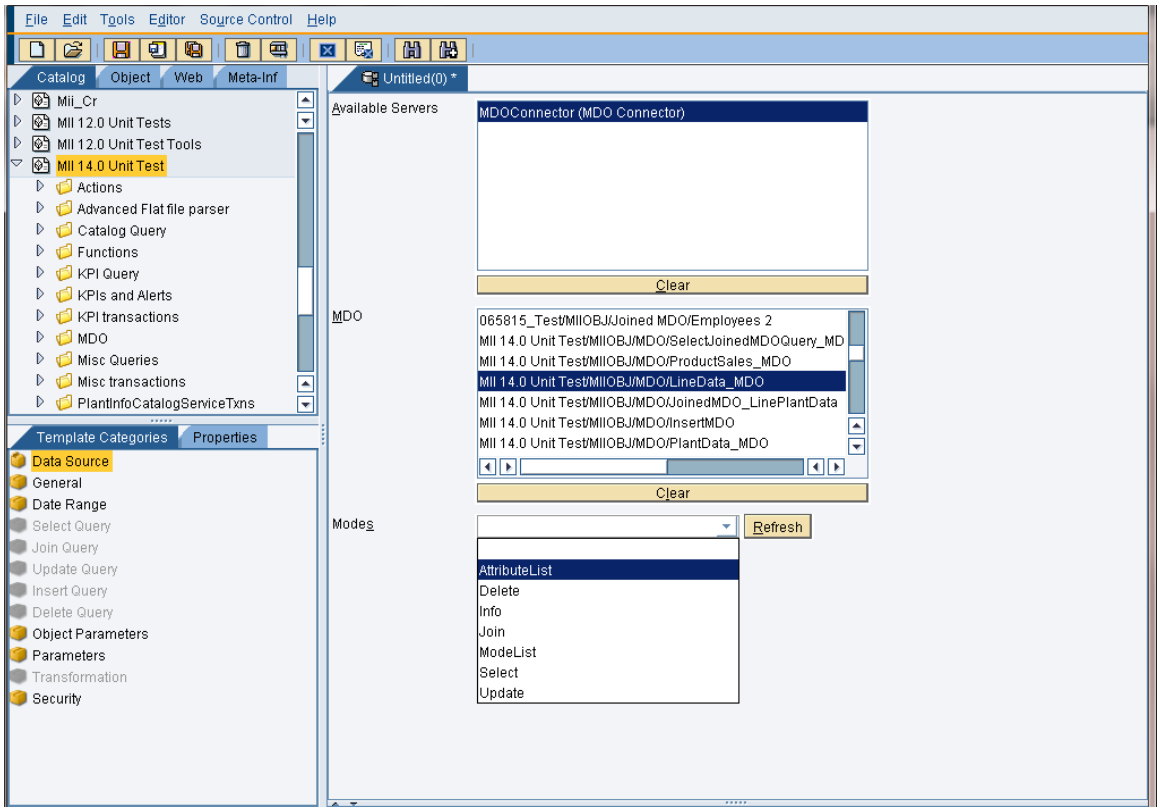
This panel has the following:-

*Available Servers* : All the MDO Connectors available in the MII system in which the MDOs are available.

*MDO*: All the MDOs available in the MII system

*Modes*: Modes supported by the selected MDO(s)

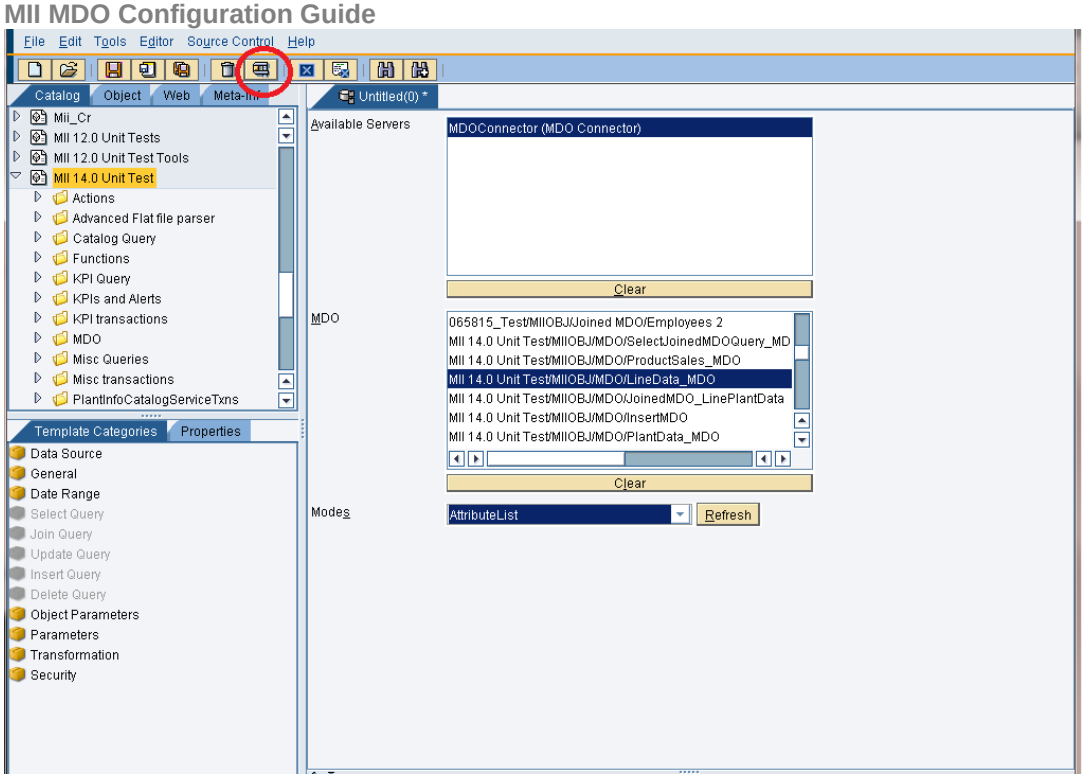
Select a MDO Connector in the available servers, Select a MDO ex: MII 14.0 Unit Test/MIIOBJ/MDO/PlantData_MDO and select any one of the modes supported

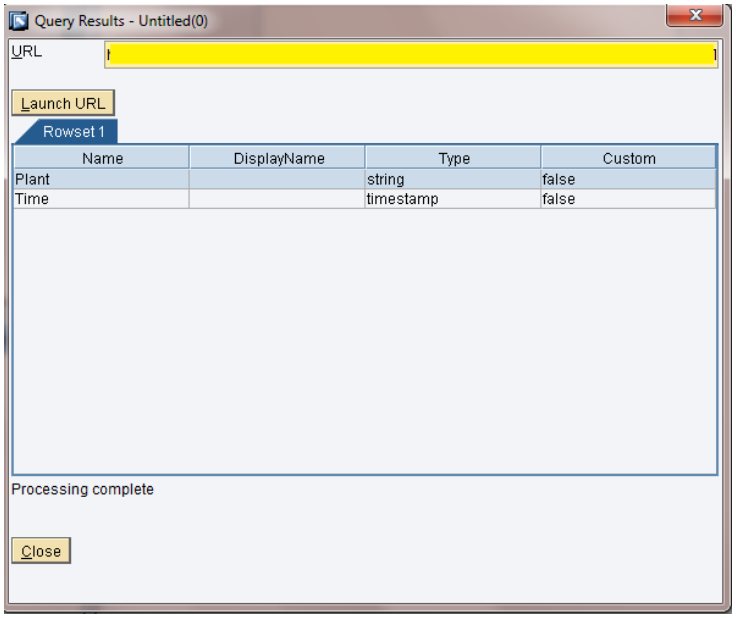



## 2.1 Execution of Command Modes

### a. AttributeList

Select the ‘AttributeList’ mode and click on Test button and then click on’OK’.



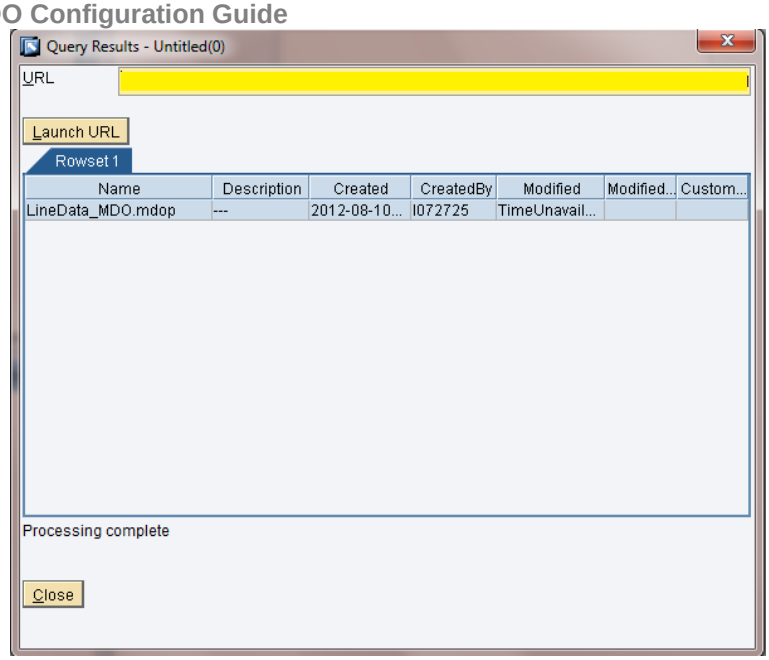




***Result :***

A list of MDO Attributes with their associated attribute configuration

### b. Info

Select the 'Info' Mode from the list of modes and click the Test button ,




***Result:***

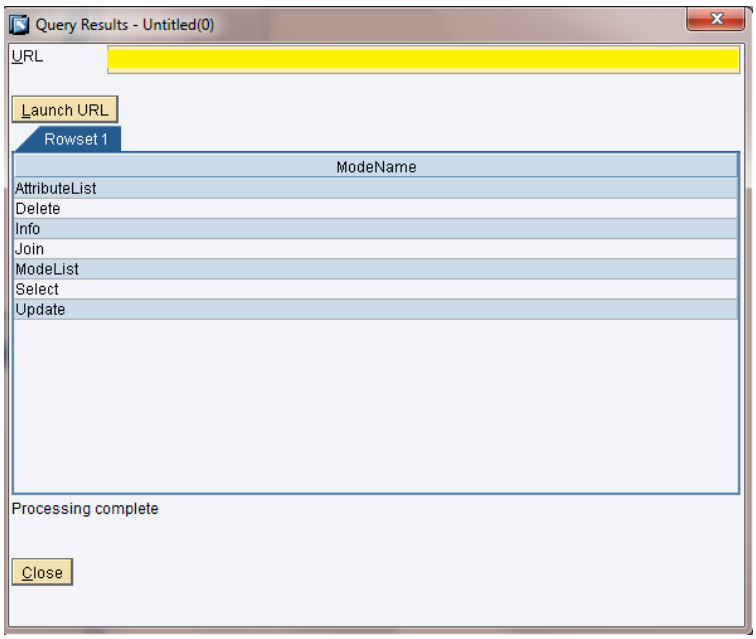
A list of the MDO Properties , including name,description,Created,Created By, Modified , Modified By and the custom Properties defined for MDO.

### c. ModeList

Select the 'ModeList' Mode from the list of modes and execute it.

***Result:***

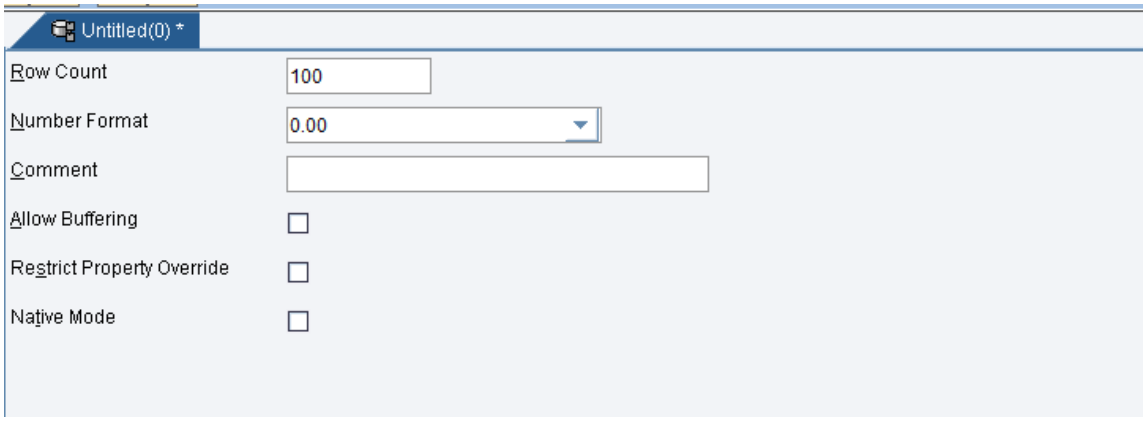
A list of modes supported by the selected MDO(s).



## 2.2 General Panel

You use the parameters on the *General* screen of an SAP Manufacturing Integration and Intelligence query to set the following:

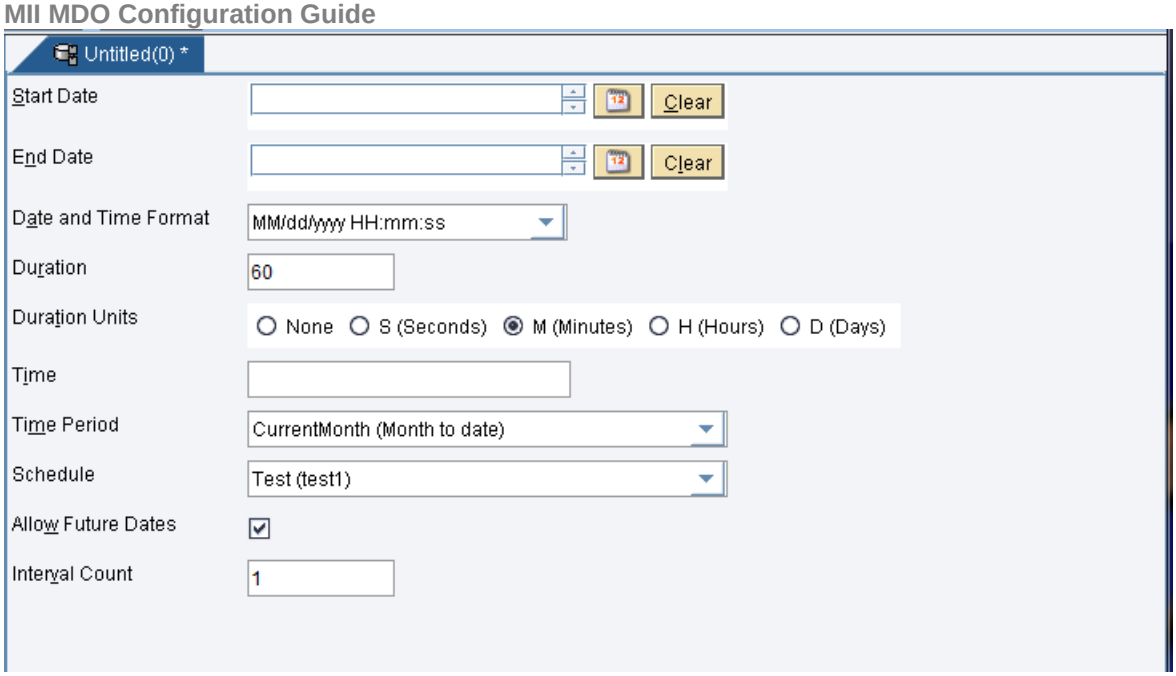
- Row Count : Maximum number of rows returned from the query
- Number Format : Numeric format for data values
- Amount of time queries are kept in a cache
- Native Mode : On clicking on the check box, it will allow the user to execute using Native SQL statements




## 2.3 Date Range

You use the *Date Range* screen to assign parameters for SAP Manufacturing Integration and Intelligence (SAP MII) time-sensitive queries

In a MDO query, the date filter is applied to the *DateColumn*, which is set on the *Join Query Details Screen.*

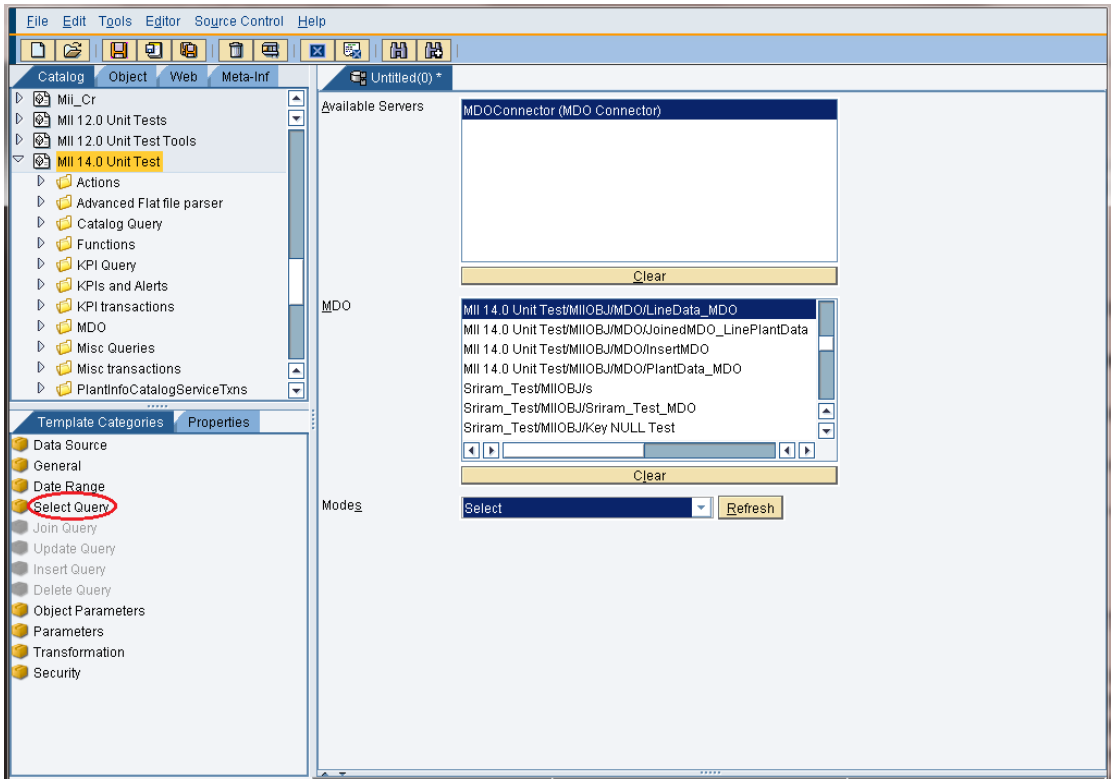



## 2.4 Select Query

If you are using *Select* mode, you use this screen to configure your query. You can select attributes from the *Available Attributes* list, which are attributes from the MDO.

Selects and returns the data for an MDO. For a persistent MDO, the persisted data is returned. For an on-demand MDO, the data provider is executed in real time and the data is returned.

**Step 1:** Go to the Data Source Panel -> Select any MDO and Select the mode Select from the list of modes. The panel in the Templat eCategories Tab is enabled.

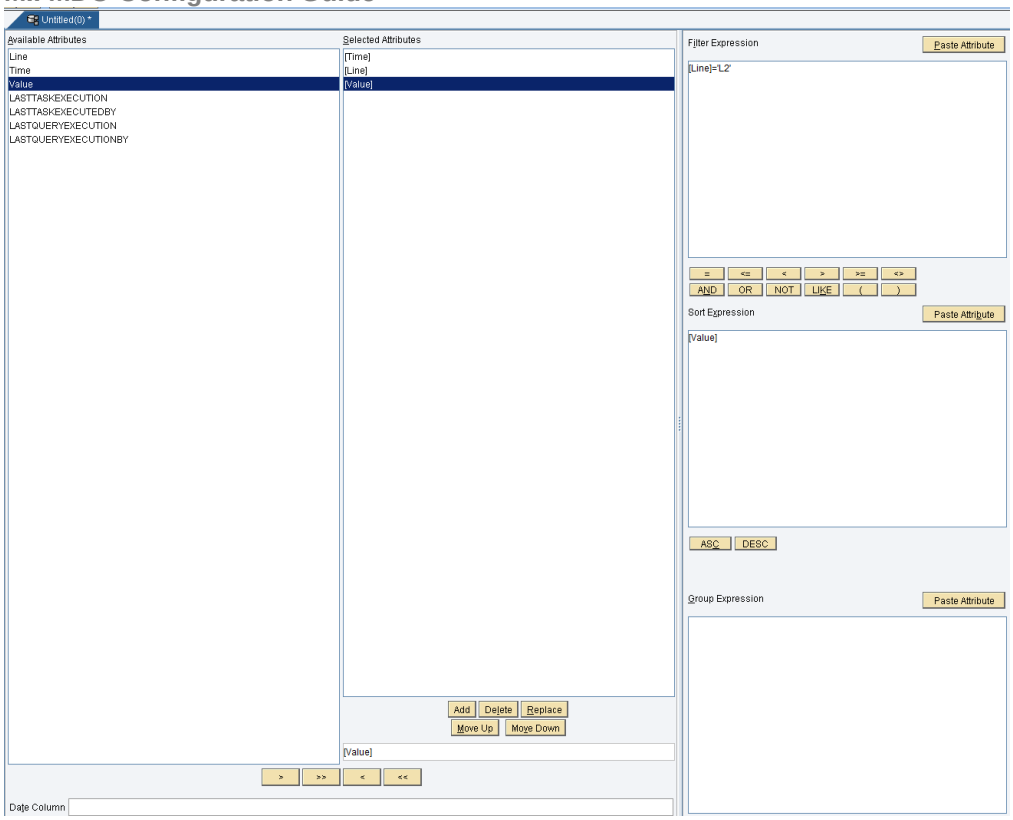



**Step 2**: Select the Select panel from tab and fill in the details in the screen.

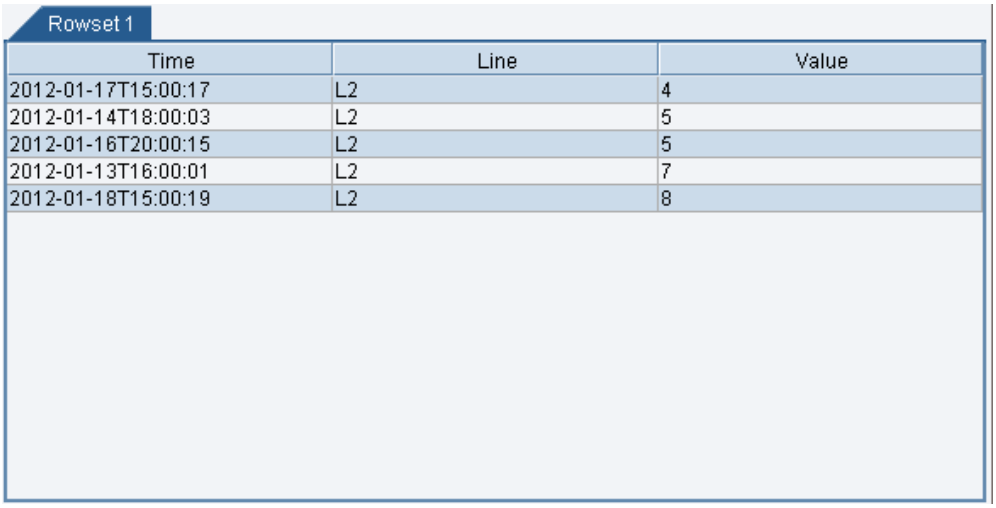
**Step 3:** Select the attributes from the ‘Available attributes ‘ column by clicking on the ‘Add button below it or the ‘>’ sign button.

‘Attributes’ list contains the attributes to select as the elements for each row

**Step 4**: Provide the Filter(where clause) , Group and Sort expression by dragging the columns from the ‘Selected Attributes’ to the respective expression areas or select any attribute in the ‘Selected Attribute’ and click on ‘Paste Attribute’




**Step 5 :** Select Test and see the results observed will be as follows. Here the the Filter and the sort has been performed on the selected attributes.

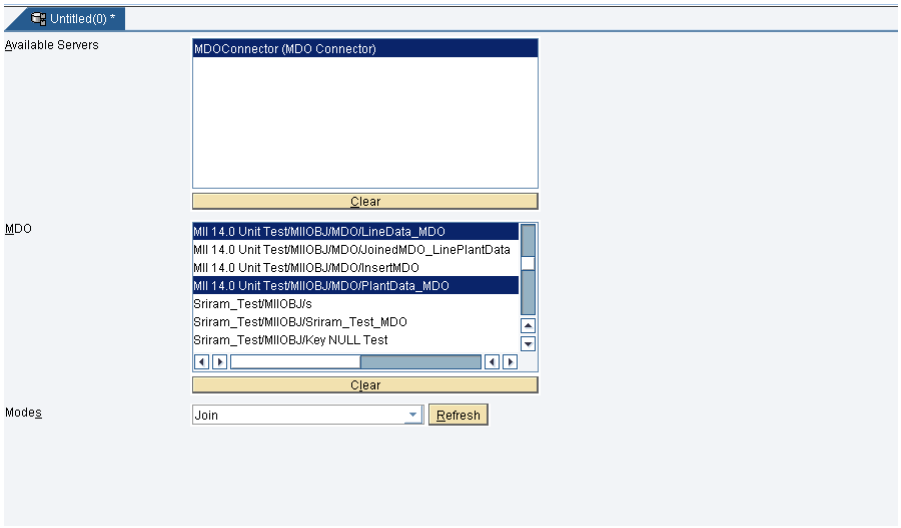



| Time | Line | Value |
|---|---|---|
| 2012-01-17T15:00:17 | L2 | 4 |
| 2012-01-14T18:00:03 | L2 | 5 |
| 2012-01-16T20:00:15 | L2 | 5 |
| 2012-01-13T16:00:01 | L2 | 7 |
| 2012-01-18T15:00:19 | L2 | 8 |

## 2.5 Join Query

A **join** is a query that combines rows from two or more tables, views. In this case we can perform joins on Persistent MDOs/Joined MDOs.

**Step 1:** Go to the Data Source Panel -> Select two Persistent MDOs/Joined MDOs and Select the mode 'Join' from the list of modes. The respective panel 'Join' in the Template Categories Tab is enabled.




Note* : Joins cannot be performed on On-Demand MDOs

**Step 2**: Go to the *Join* panel and perform the following.

i. Select a MDO from the '*Selected Objects'* list which populates the MDOs selected on the Data Source Panel and below it displays the attributes of the MDO

ii. Select the attributes from the '*Available Attributes'* list .The selected attribute appears in the TextField below it. Click on 'Add'/drag to add the attribute or 'Add All' to add all the attributes in the '*Available Attributes'* list to the '*Selected Attributes'* table.

Example : Select the MDO *MII 14.0 Unit Test/MIIOBJ/MDO/PlantData_MDO* from '*Selected Objects' and* select the attributes 'Time' and 'Value' from the '*Available Attributes'* list .Click on 'Add' button to add the selected attributes in the '*Selected Attributes'* table. Then select the second MDO *MII 14.0 Unit Test/MIIOBJ/MDO/LineData_MDO* from '*Selected Objects' and* select the attribute 'Plant' from the '*Available Attributes'* list .Click on 'Add' button to add the selected attributes in the '*Selected Attributes'* table.

Note*: The '*Selected Attributes'* table consists of two columns namely:

*Selected Attributes* : Contains which attributes to select as the elements for each row. It contains the entire path location of the MDO and its attribute. This is non-editable field. Here in this example the selected Attributes are the entire paths of the respective attributes like *MII 14.0 Unit Test/MIIOBJ/MDO/LineData_MDO/Line,*

*MII 14.0 Unit Test/MIIOBJ/MDO/PlantData_MDO/Time*

and *MII 14.0 Unit Test/MIIOBJ/MDO/PlantData_MDO/Value*

*Display Name:* It has the alias name of the selected attribute(editable field).When the query is executed , the aliases for the selected attributes will be displayed as the columns. Here in this example the display names are *Plant , PlantTime and PlantValue.*

iii. You can change the Display Names of the selected Attributes by double clicking on the table row of the Display Name and it becomes editable.

iv. To add the Join Expression ,Click on 'Add' button / usage of the Secondary mouse click and select 'Add' from the context menu.

v. Select an MDO from the first column that represents the first table say *MII 14.0 Unit Test/MIIOBJ/MDO/LineData_MDO*. Select the drop down of join type and select a join(Inner Join and Left Outer Join) say for this example *INNER JOIN*. Similarly select the MDO from the dropdown of the third table column say *MII 14.0 Unit Test/MIIOBJ/MDO/PlantData_MDO*

vi. To add a condition to the respective Join. Select the join(in the '*Joins'* Table).Click on the 'Add' button /Secondary mouse click and select 'Add' from the context menu in the 'C*onditions'* Table and it adds a row in the respective table.

vii. Select the first column which contains all the attributes of the MDO selected in the first column of the *Joins* table say '*Time'*. Select the operator say '>'and an attribute in the third column from the list which contains the attributes of the MDO selected in the third column of the *Joins* table say '*PlantTime'*.

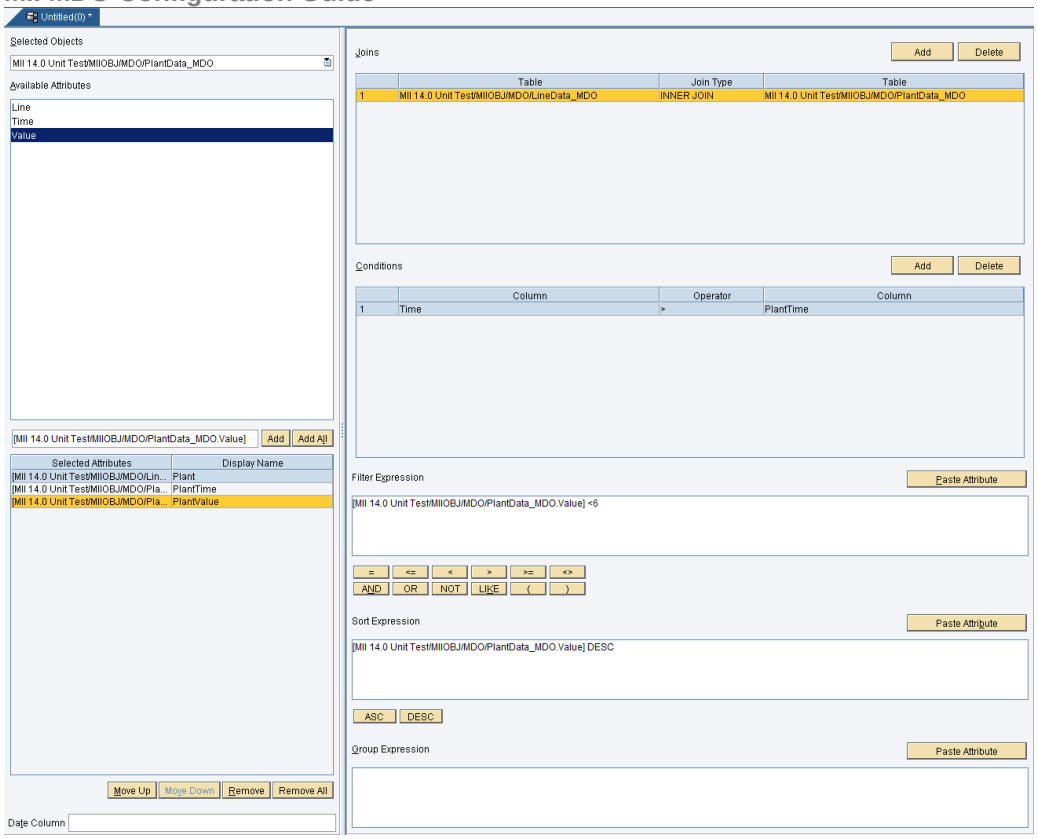
viii. Provide the Filter ,Sort and Group expression by dragging the attributes /selecting the attribute from the '*Available Attributes'* or the '*Selected Attributes'* table and clicking on '*Paste Attribute'* button.

Hence say you select the third attribute from the '*Selected Attributes'* table namely *PlantValue* and click on '*Paste Attribute'* button provided with the Filter area and give the expression followed by the pasted attribute as '> 6' hence the filter expression is

'*MII 14.0 Unit Test/MIIOBJ/MDO/PlantData_MDO/PlantValue>6'*

Now select the second attribute from the '*Selected Attributes'* table namely *PlantValue* and click on '*Paste Attribute'* button provided with the Sort area and give the expression followed by the pasted attribute as 'DESC' to sort in the descending order, hence the sort expression is

'*MII 14.0 Unit Test/MIIOBJ/MDO/PlantData_MDO/PlantValue DESC'*




**Step 9**: To test the MDO query, choose *Test*

***Result:***

It returns the result which retrieves the joins performed on 'Line_Data' MDO and'Plant_Data' MDO of the plants with name 'P1' (filter expression). And the result is sorted in the Descending order based on the 'PlantValue'

Rowset 1

| PLANT | PLANTTIME | PLANTVALUE |
|---|---|---|
| P1 | 2012-01-16T20:00:14 | 5 |
| P1 | 2012-01-16T20:00:14 | 5 |
| P1 | 2012-01-16T20:00:14 | 5 |
| P1 | 2012-01-16T20:00:14 | 5 |
| P1 | 2012-01-16T20:00:14 | 5 |
| P1 | 2012-01-16T20:00:14 | 5 |
| P1 | 2012-01-17T16:00:18 | 5 |
| P1 | 2012-01-14T18:00:03 | 5 |
| P1 | 2012-01-14T18:00:03 | 5 |
| P1 | 2012-01-14T18:00:03 | 5 |
| P1 | 2012-01-14T18:00:03 | 5 |
| P1 | 2012-01-14T18:00:03 | 5 |
| P1 | 2012-01-14T18:00:03 | 5 |
| P1 | 2012-01-14T18:00:03 | 5 |
| P1 | 2012-01-14T18:00:03 | 5 |
| P1 | 2012-01-14T18:00:03 | 5 |

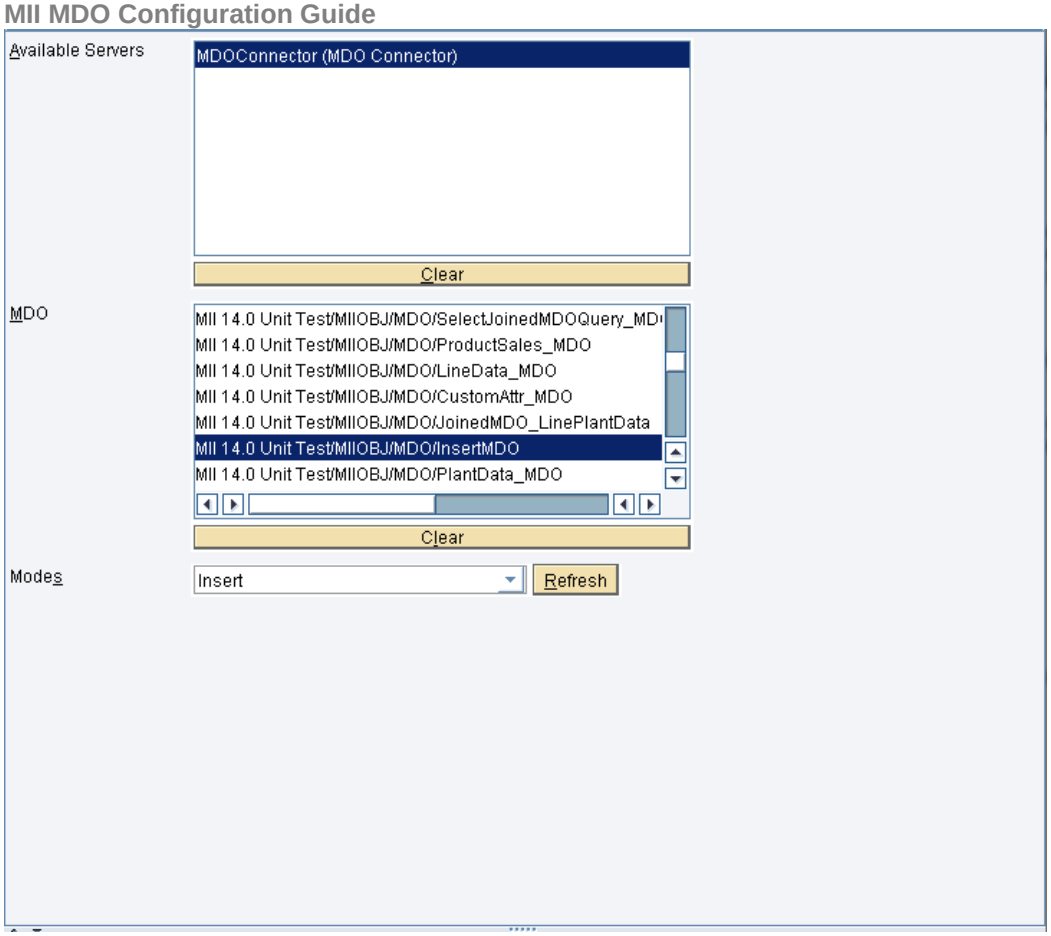
## 2.6 Insert Query

You use this screen to configure your query.

This mode allows the user to specify the attribute values used when a new row of data is stored in the MDO Data Results..

**Step 1:** Go to the Data Source Panel -> Select a MDO(Persistent or On-Demand MDO).

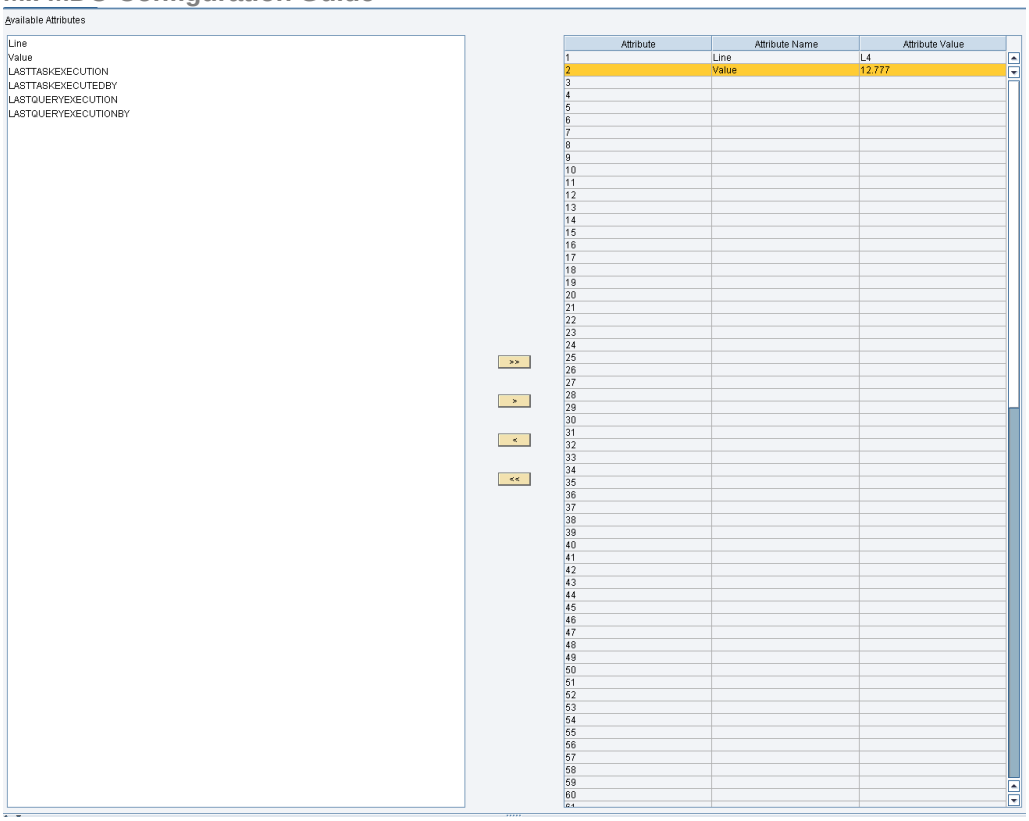
Note* : The ***Insert Mode*** is available in the list of modes only when the MDO Object has custom attributes.




**Step2:** Select the mode 'Insert' from the list of modes. The respective panel 'Insert Query' in the Template Categories Tab is enabled.

**Step 3:** In the 'Insert Query ' Panel ,add the attributes by selecting the attributes listed in the *Available Attributes* list and clicking on '>' or'>>' to add all the custom attributes.

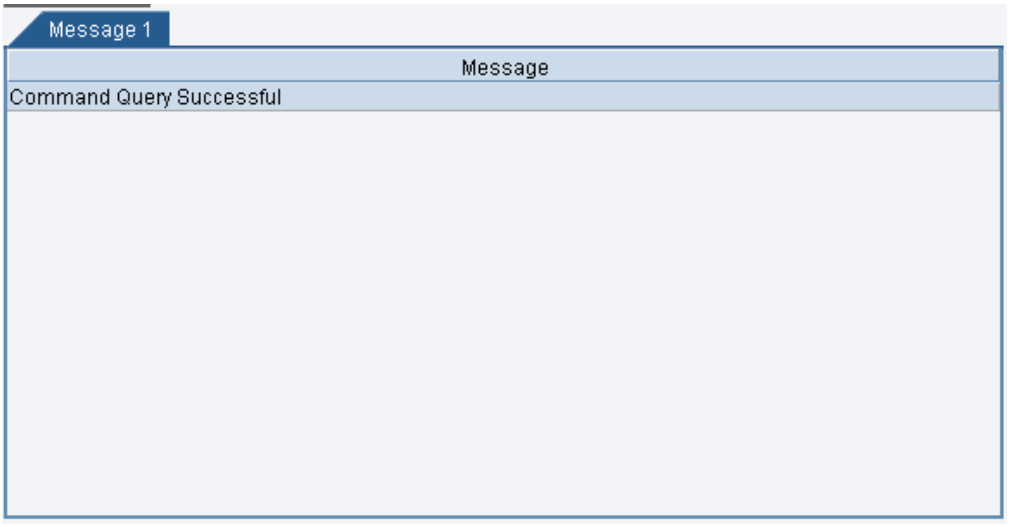
**Step 4:** Fill in the values in the 'AttributeValue' column which represent the respective values of the attribute.




**Step 5:** To test the MDO query, click *Test*

***Result:***

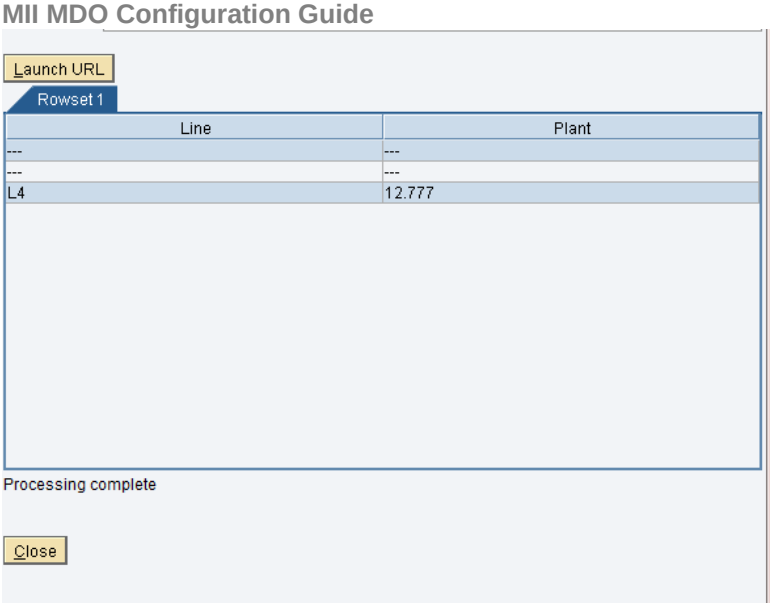
It displays the Success message 'Command Query Successful'.




To check the row entry in the selected MDO, Execute the MDO

***Result***

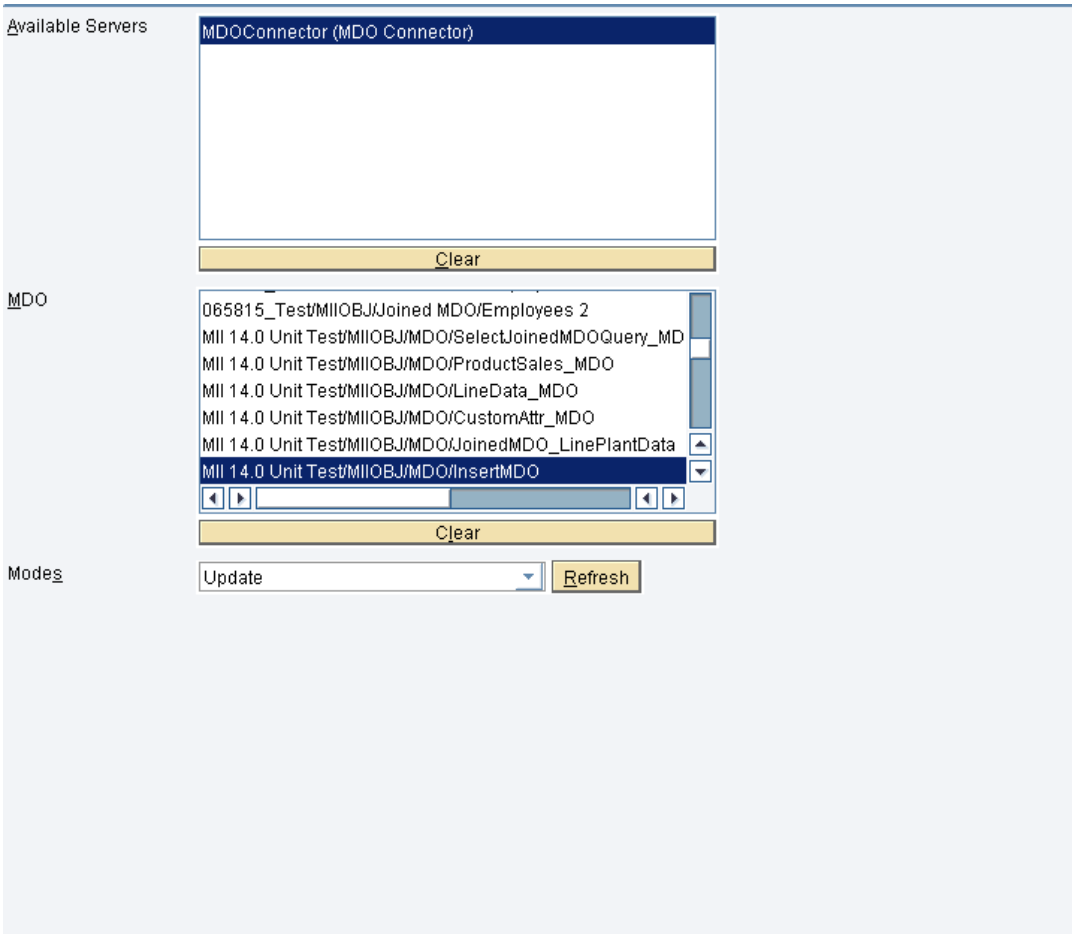
It creates a new row of data with the Line and Value details.




## 2.7 Update Query

The Update Query is used to update the persisted data for an MDO with Custom Attributes.

**Step 1:** Go to the Data Source Panel -> Select a MDO(Persistent).

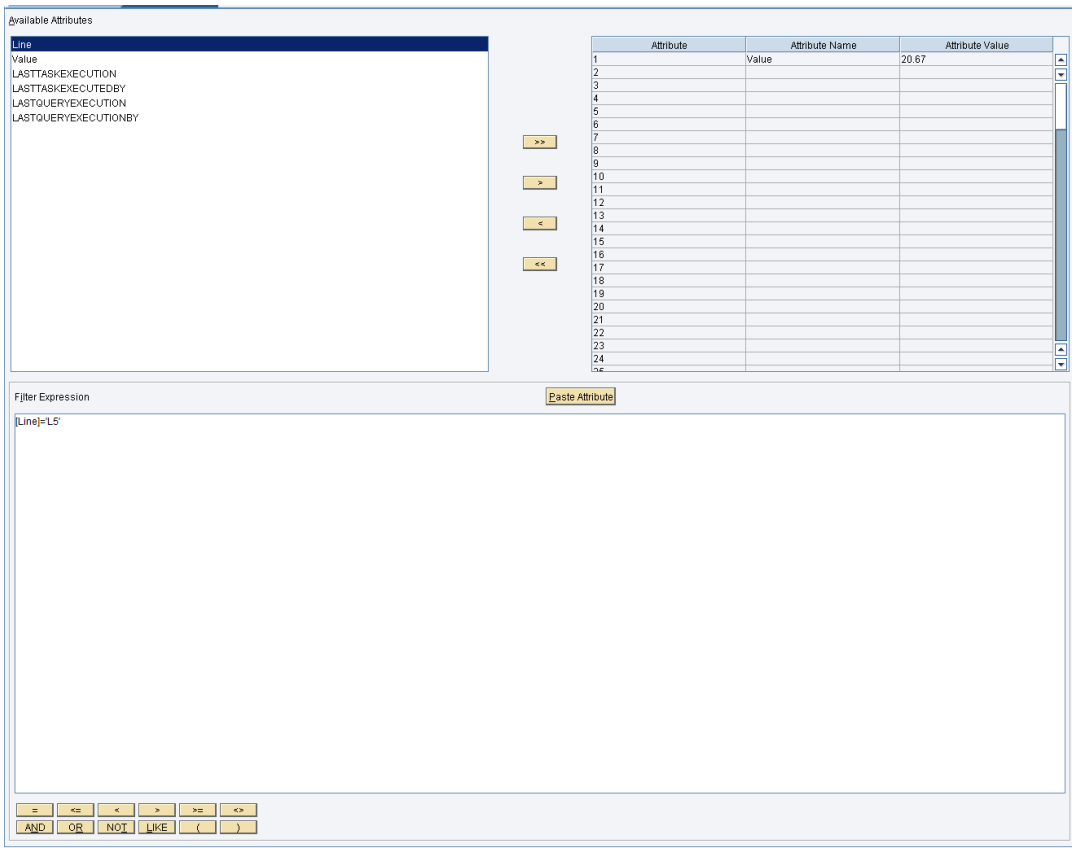


**Step2:** Select the mode 'Update' from the list of modes. The respective panel 'Update Query' in the Template Categories Tab is enabled.

**Step 3:** In the 'Update Query ' Panel ,add the attributes by selecting the attributes listed in the *Available Attributes* list and clicking on '>' or'>>' to add all the custom attributes.

**Step 4:** Fill in the values in the 'Attribute Value' column which represent the respective values of the attribute provided in the 'Attribute Name' column.
Example :Enter the value for the 'attribute Name' =" Value" as 20.47

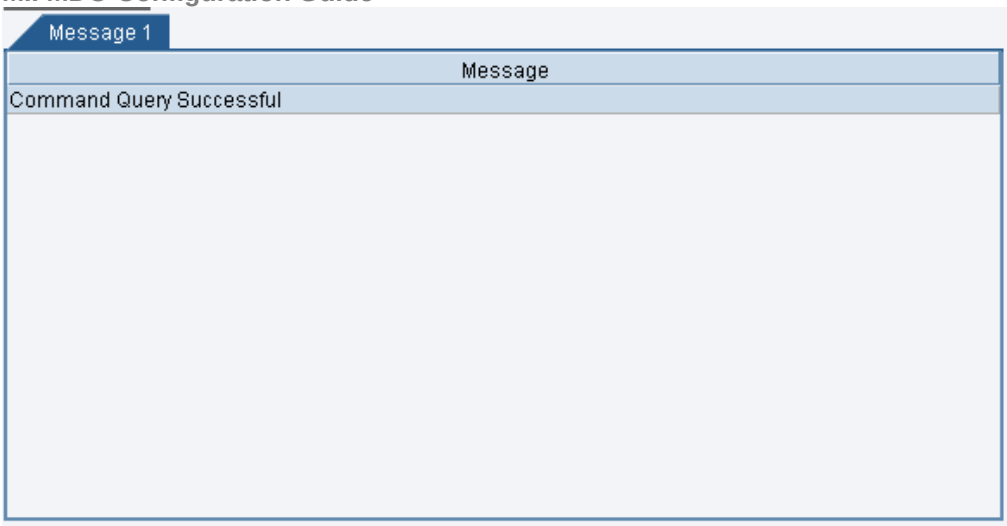
**Step 5:** Provide the input for the 'Filter Expression'
Example :And the Filter expression (where clause) as [Line] = 'L5'




**Step 6:** To test the MDO query, click *Test*

***Result:***

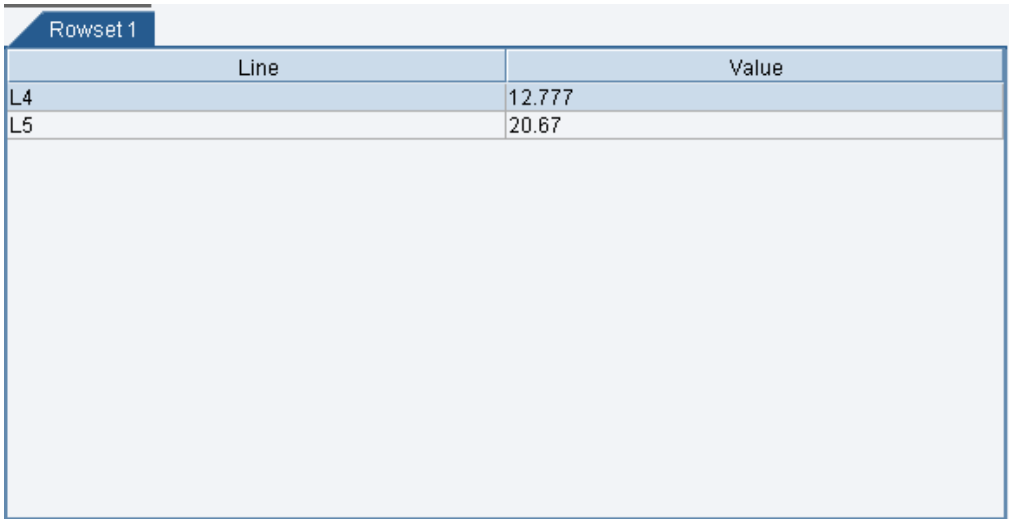
It displays the Success message 'Command Query Successful'.




To check the row entry in the selected MDO, Execute the MDO

***Result***

It updates the Value of the Line of name 'L5' to '20.67'.



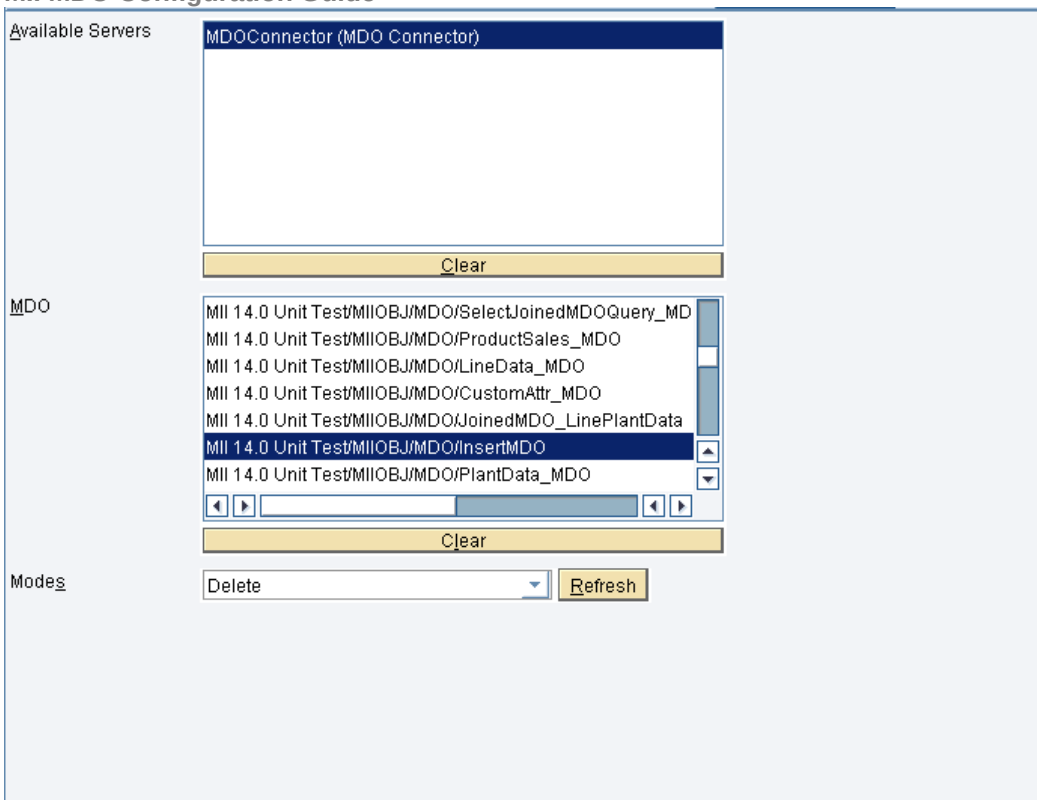

## 2.8 Delete Query

The attributes from the MDO are listed in the *Available Attributes* list.

You can use the *Filter Expression* to update specific records in the persisted data. If you do not provide a filter statement, all records in the persisted data are deleted.

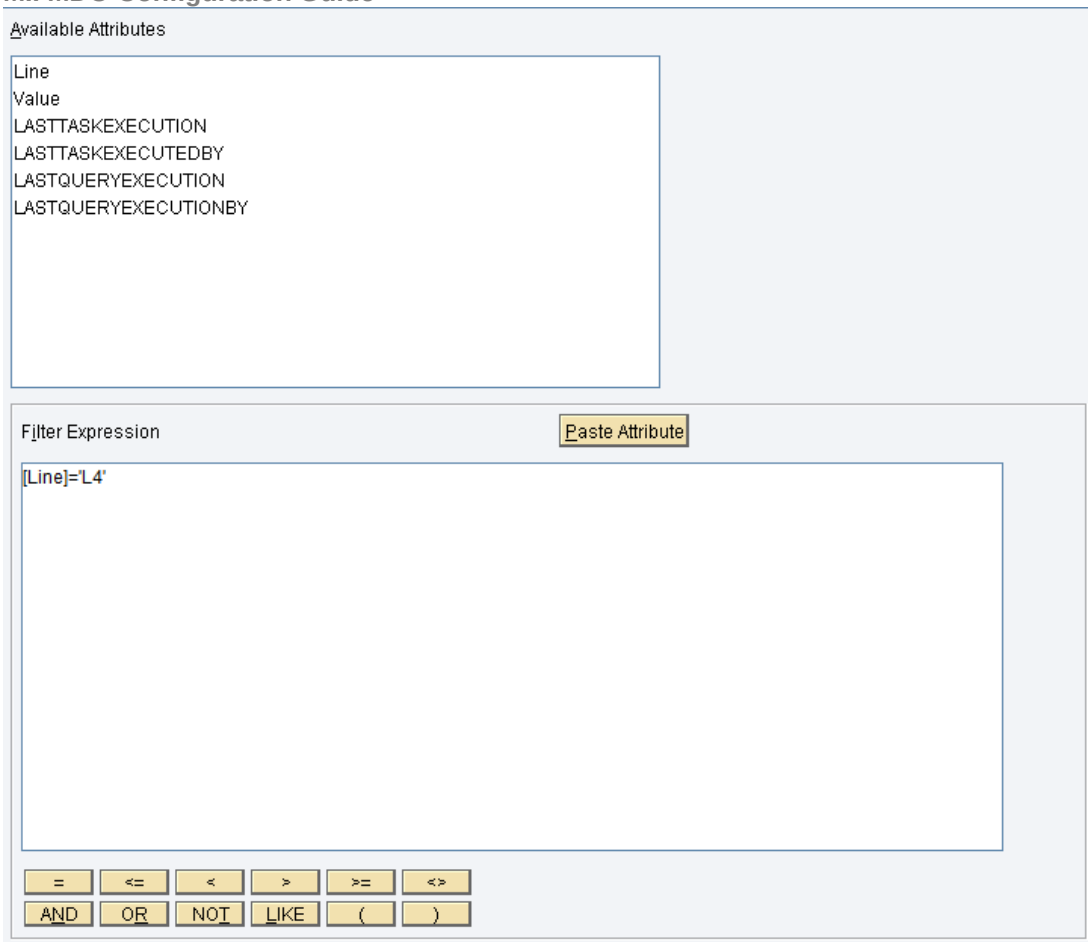
The Delete Query is used to delete the persisted data for a MDO (Persistent MDO) with Custom Attributes.

**Step 1:** Go to the Data Source Panel -> Select a MDO(Persistent).




**Step2:** Select the mode 'Delete' from the list of modes. The respective panel 'Delete Query' in the Template Categories Tab is enabled.

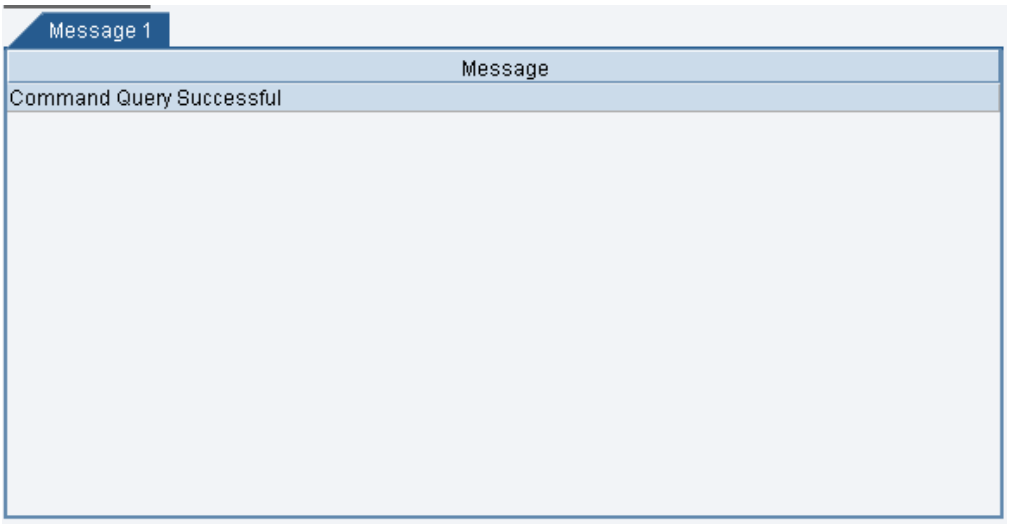
**Step 3:** In the 'Delete Query ' Panel , Provide the input for the 'Filter Expression'
Example :And the Filter expression (where clause) as [Line] = 'L4'




**Step 4:** To test the MDO query, click *Test*

***Result:***

It displays the Success message 'Command Query Successful'.



To check the row entry in the selected MDO, Execute the MDO

***Result***

It deletes the entry based on the Filter expression defined where Line of name 'L4'.

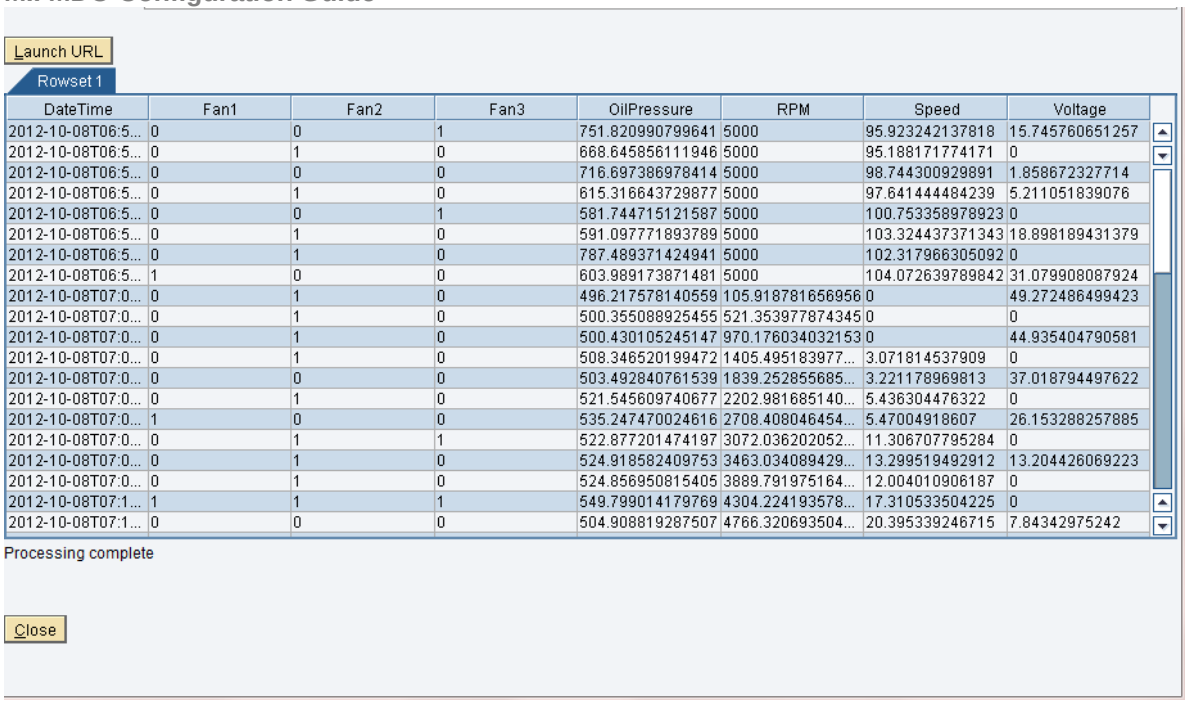
Rowset 1

| Line | Value |
|---|---|
| L5 | 20.67 |

## 2.9 Object Parameters

Object parameters are passed from the MDO query to an On-Demand MDO. These can then be linked to the dynamic query parameters of the MDO definition, which can then be passed into the MDO data provider. This enables the user to modify the data provider execution at runtime.

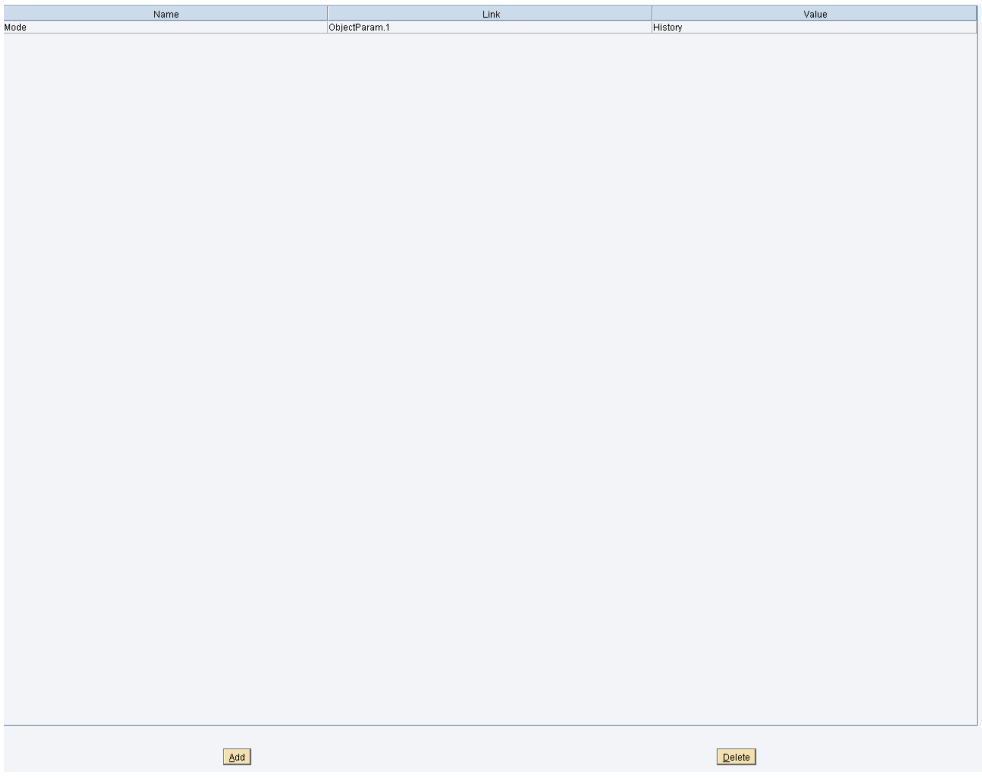
**Step 1**: Consider an On-Demand MDO Object which has a Tag Query as the Data Provider which has the '*History*' mode of getting all the details of Engine Fans with their respective OilPressure , RPM Speed and Voltage values.



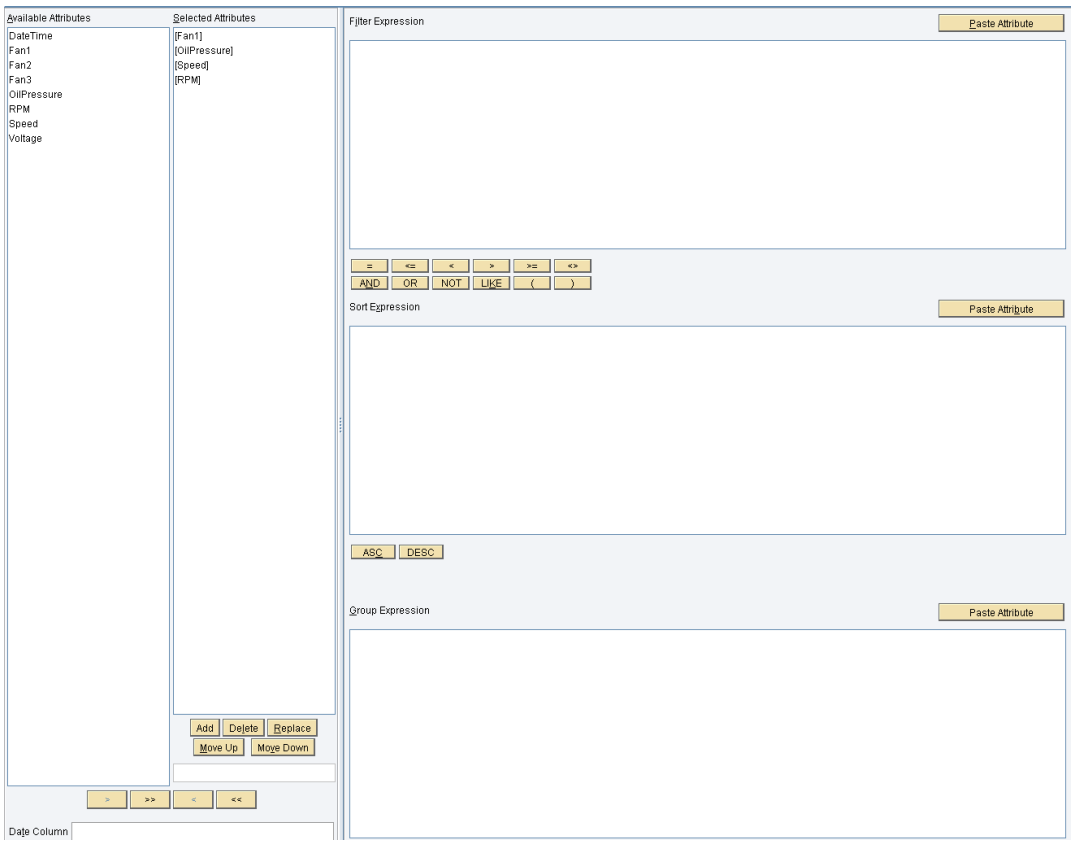

| DateTime | Fan1 | Fan2 | Fan3 | OilPressure | RPM | Speed | Voltage |
|---|---|---|---|---|---|---|---|
| 2012-10-08T06:5... | 0 | 0 | 1 | 751.820990799641 | 5000 | 95.923242137818 | 15.745760651257 |
| 2012-10-08T06:5... | 0 | 1 | 0 | 668.645856111946 | 5000 | 95.188171774171 | 0 |
| 2012-10-08T06:5... | 0 | 0 | 0 | 716.697386978414 | 5000 | 98.744300929891 | 1.858672327714 |
| 2012-10-08T06:5... | 0 | 1 | 0 | 615.316643729877 | 5000 | 97.641444484239 | 5.211051839076 |
| 2012-10-08T06:5... | 0 | 0 | 1 | 581.744715121587 | 5000 | 100.753358978923 | 0 |
| 2012-10-08T06:5... | 0 | 1 | 0 | 591.097771893789 | 5000 | 103.324437371343 | 18.898189431379 |
| 2012-10-08T06:5... | 0 | 1 | 0 | 787.489371424941 | 5000 | 102.317966305092 | 0 |
| 2012-10-08T06:5... | 1 | 0 | 0 | 603.989173871481 | 5000 | 104.072639789842 | 31.079908087924 |
| 2012-10-08T07:0... | 0 | 1 | 0 | 496.217578140559 | 105.918781656956 | 0 | 49.272486499423 |
| 2012-10-08T07:0... | 0 | 1 | 0 | 500.355088925455 | 521.353977874345 | 0 | 0 |
| 2012-10-08T07:0... | 0 | 1 | 0 | 500.430105245147 | 970.176034032153 | 0 | 44.935404790581 |
| 2012-10-08T07:0... | 0 | 1 | 0 | 508.346520199472 | 1405.495183977... | 3.071814537909 | 0 |
| 2012-10-08T07:0... | 0 | 0 | 0 | 503.492840761539 | 1839.252855685... | 3.221178969813 | 37.018794497622 |
| 2012-10-08T07:0... | 0 | 1 | 0 | 521.545609740677 | 2202.981685140... | 5.436304476322 | 0 |
| 2012-10-08T07:0... | 1 | 0 | 0 | 535.247470024616 | 2708.408046454... | 5.47004918607 | 26.153288257885 |
| 2012-10-08T07:0... | 0 | 1 | 1 | 522.877201474197 | 3072.036202052... | 11.306707795284 | 0 |
| 2012-10-08T07:0... | 0 | 1 | 0 | 524.918582409753 | 3463.034089429... | 13.299519492912 | 13.204426069223 |
| 2012-10-08T07:0... | 0 | 1 | 0 | 524.856950815405 | 3889.791975164... | 12.004010906187 | 0 |
| 2012-10-08T07:1... | 1 | 1 | 1 | 549.799014179769 | 4304.224193578... | 17.310533504225 | 0 |
| 2012-10-08T07:1... | 0 | 0 | 0 | 504.908819287507 | 4766.320693504... | 20.395339246715 | 7.84342975242 |

**Step 2:** Go to the *'Dynamic Query Parameters'* screen and add a row .A popup appears which is the Parameter Dialog box. Select 'Mode' in the '*Name'* column and further select 'ObjectParam.1' from the dropdown list in the '*Link'* column. In the '*Value'* column it displays the mode in which the data provider is running on(ex: "History").

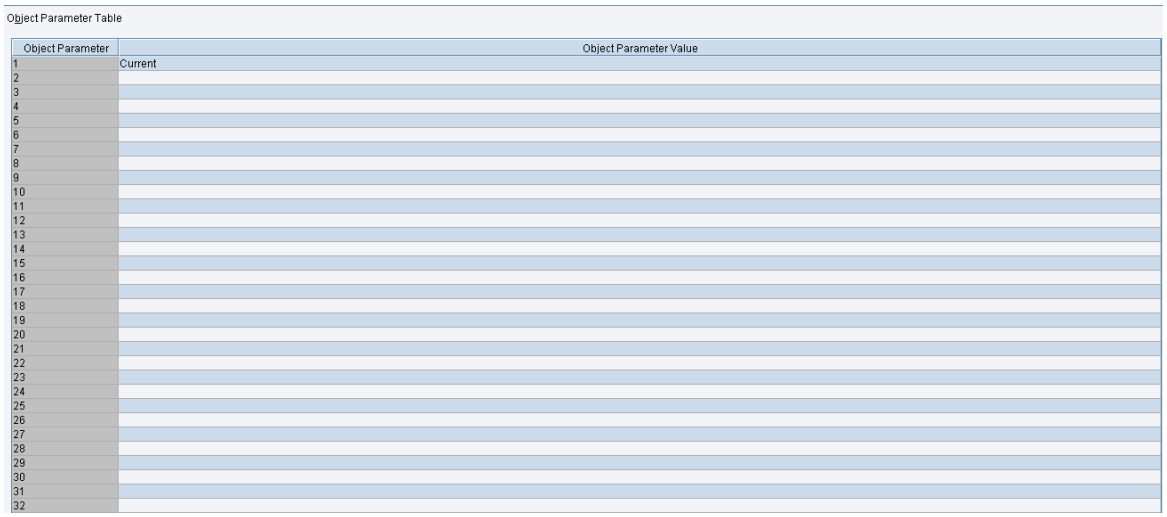
Now save the On-Demand MDO Object.



**Step 3:** Consider a MDO Select Query which has the previously created On-Demand MDO and selects few attributes to perform the query.



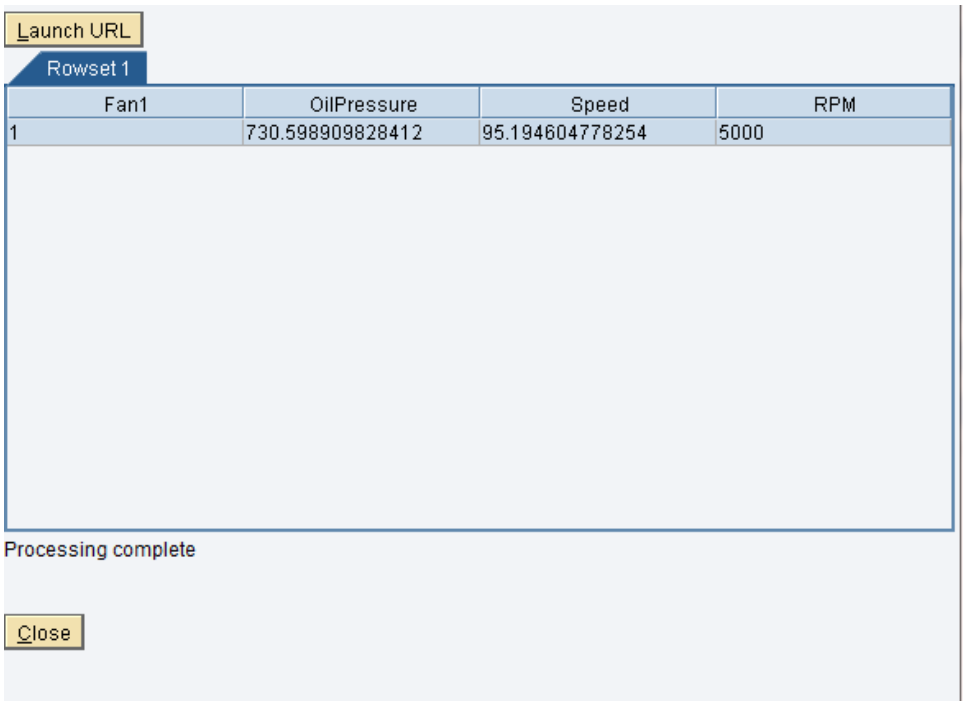

**Step 3:** Go to the Object Parameters panel of the MDO Query created previously and provide an Object parameter Value ex: “Current”



**Step 4:** To test the MDO query, click *Test* and observe the results

***Result:***

It dynamically changes the mode which was 'History' to 'Current' and hence it retrieves the data in the 'Current' Mode. Hence it enables the user to modify the data provider execution at runtime.




## 2.10 Parameters

You use this screen to specify the value for parameters defined in the MDO query attribute values or filter statement.

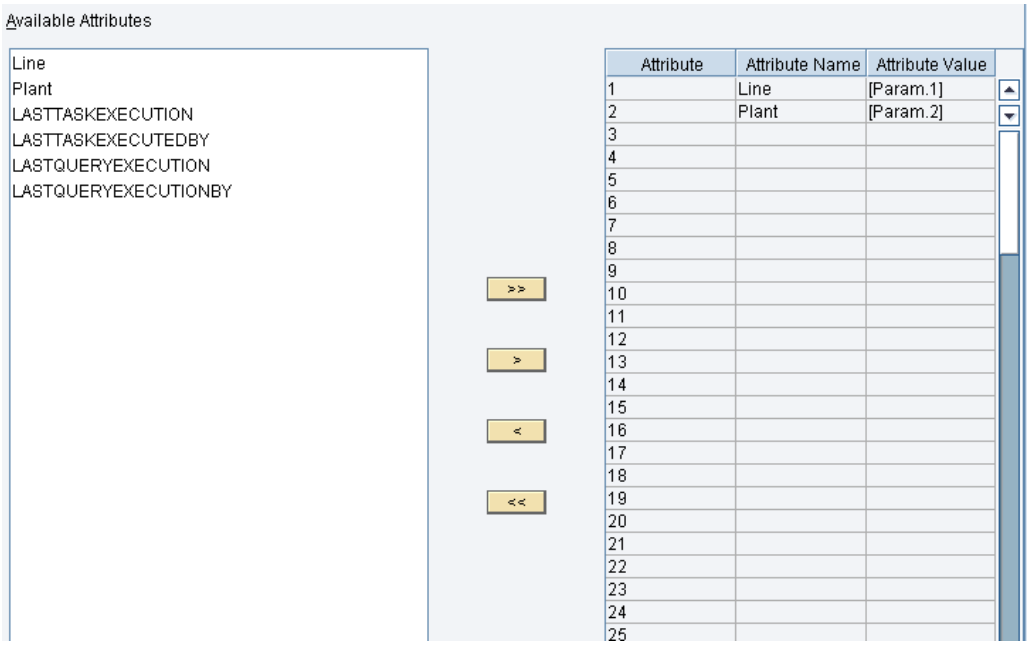
Rather than using the values in the Filter, Sort or Group expressions, we can refer to the parameters like [Param.1],[Param.2],... and so on where they respectively specify the values in the Parameters table in the 'Parameters 'Panel in the same order.

### Parameter reference

You can use parameter reference in all the MDO Query modes. Let us consider one mode: Insert Query

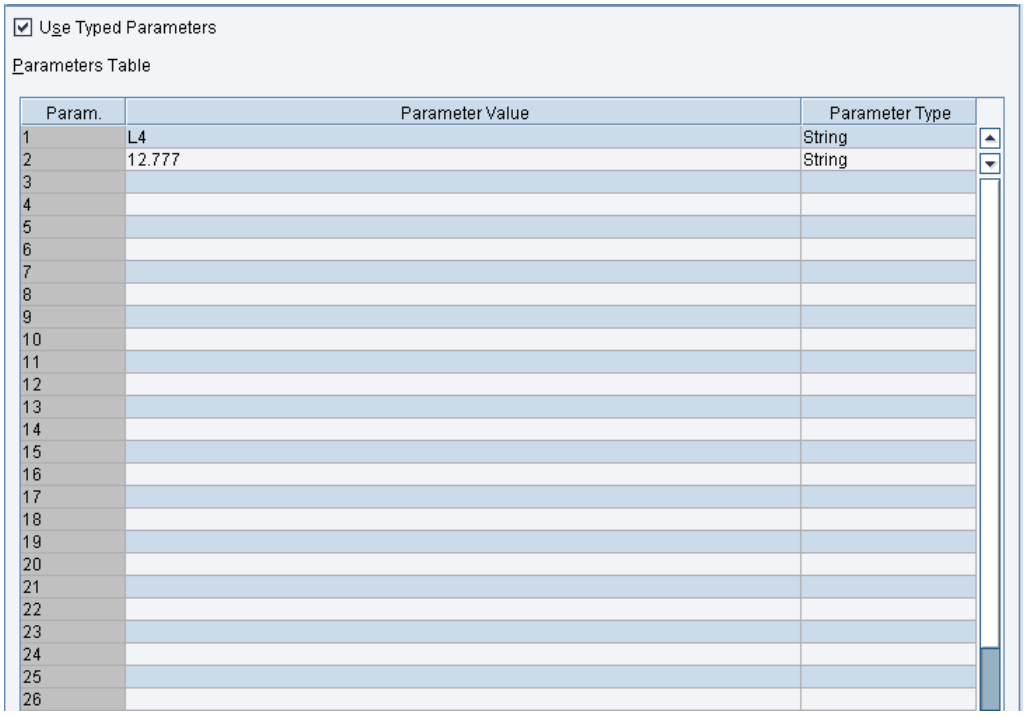
**Step 1:** Follow the steps1-3 of Insert Query

**Step 2:** Replace the values in the '*Attribute Values'* column with [Param.1] and[Param.2] respectively.




**Step 3:** Go to the '*Parameters'* screen and enter the values (in the first two entries referring to Param.1 and Param.2)

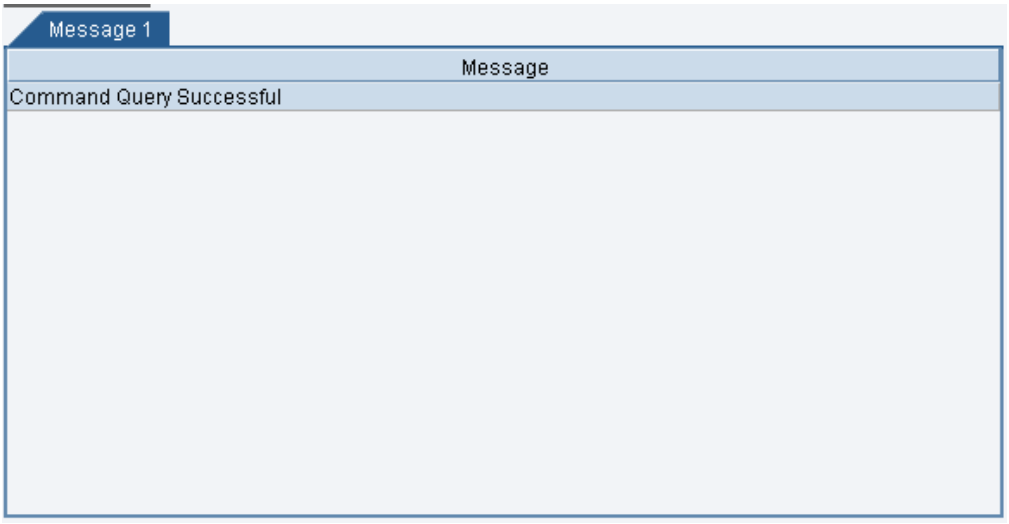
**Step 4:** Use the *Parameter Type* dropdown to select the data type for the parameter. The system checks the parameter value against the selected data type.




**Step 5:** To test the MDO query, click *Test*

***Result:***

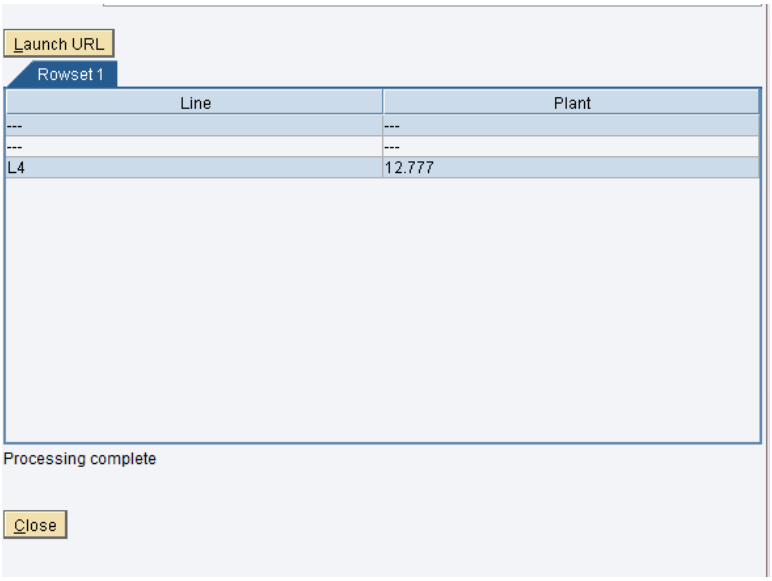
It displays the Success message 'Command Query Successful'.




To check the row entry in the selected MDO, Execute the MDO

***Result***

It creates a new row of data with the Line and Value details.




## 2.11 Transformation

You use the *Transformation* screen for queries to define an SAP MII inline transform and the parameters used by the transform.

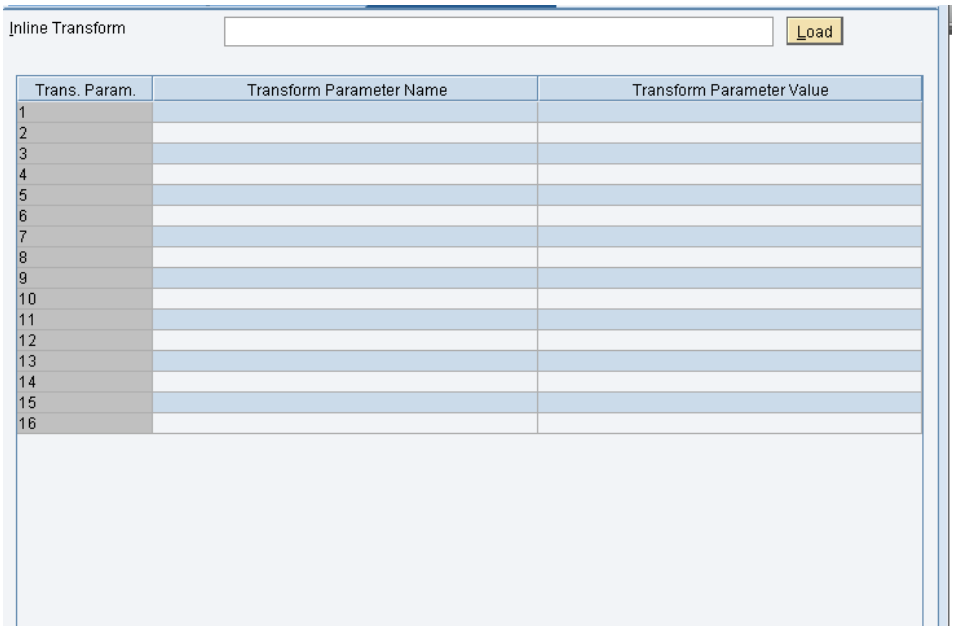
You use the inline transformation to provide additional reshaping, aggregation, or other manipulation of a data set before it is rendered. A number of inline transformations are provided with the SAP MII installation which can be used to provide mechanisms for reshaping or performing common data transformation tasks on XML

data sets being generated from SAP MII queries. The inline transform is used to perform query-specific manipulation of returning data

The inline transformations are not directly responsible for controlling the visual output of a query but are used to alter the data content by converting it from one SAP MII XML data set to another. The resultant data set can then be used by any other SAP MII component like applets, reports, and so on.

The following are a few examples describing the inline transform:

- Calculate a range between values
- Calculate a running sum for a column
- Providing data summaries by inputting

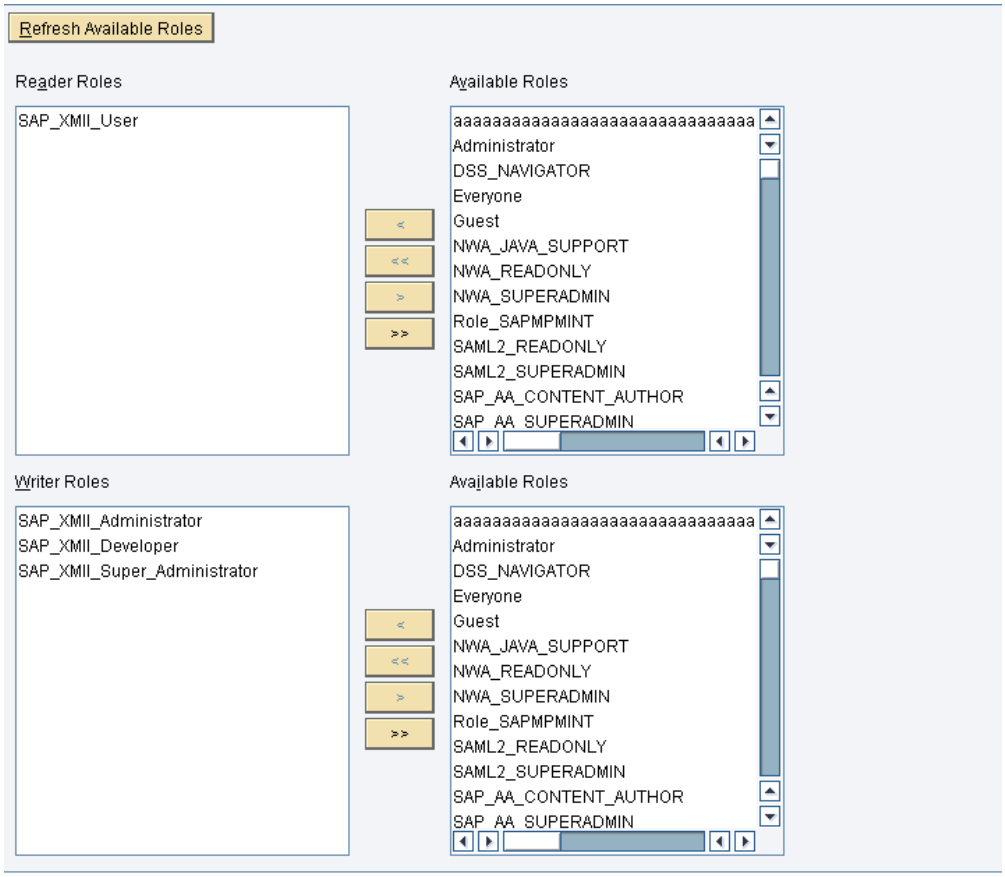



## 2.12 Security

You use this screen to authorize roles to read or write to a MDO Query.

You can select the roles from the '*Available Roles'* column and add it to the '*Reader Roles'* and '*Writer Roles'* by clicking on '<' or'<<' to select all.

Hence only the roles specified in the '*Reader Roles'* and '*Writer Roles'* will have the respective authorizations to read or write respectively to a MDO Query.




# 3. Appendix

## 3.1 Icon Library

The list below shows all the different icons representing the MDOs .

| | |
|---|---|
|  | Persistent MDO |
|  | On-Demand MDO |
|  | Joined MDO |
|  | MDO Query |

## 3.2 Frequently Asked Questions

1. **Is "Persistent MDO" the best approach for temporary data storage in MII?**

   **If so, how can I populate MDO with data?**

   Yes Persistent MDO is the best approach to store data in MII. To insert the data into the MDO, use Insert Query. Make sure that you have used you have used Param.1, Param.2 etc in MDO query to accept the input parameters.

2. **Why has the MDO Query OpenSQL Compliant?**

   Open SQL allows you to access all database tables known to the SAP system where as the Native SQl allows only database-specific SQL statements. For more information on the OpenSQl Grammer :

   http://help.sap.com/saphelp_nw70/helpdata/en/9b/f46cabaa874bc9a82234e8cf1d0696/content.htm

   All programming models provided by SAP for database access are based on the Open SQL for Java framework. This provides performance-enhancing mechanisms, such as table buffering and statement pooling, while providing vendor independent access to various databases, such as Oracle, IBM DB2, Microsoft SQL Server, and the SAP-certified open source database MaxDB. Applications do not need to be changed because Open SQL works in all of the supported databases.

   The Open SQL for Java framework provides a wrapper for the proprietary JDBC drivers. In doing so, it provides the binding layer between AS Java and proprietary JDBC drivers at runtime. This ensures efficient and portable SQL-based data selection and manipulation.

3. **Why only two types of joins namely: Inner Join and Left Outer Join? And Why not Right Outer Join?**

   OpenSQL supports only these two joins and as you know MDO Query is now OpenSQL Compliant.

4. **Is there any setting required for executing Native SQL statements in the MDO Query?**

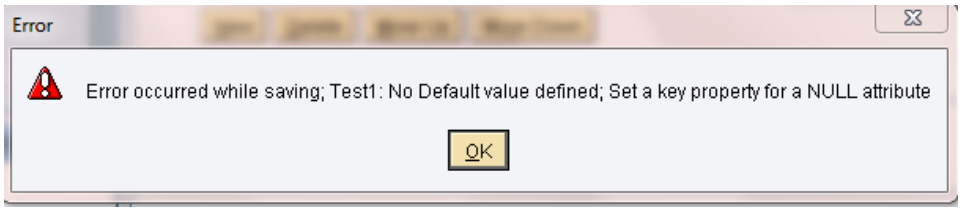
   In the *General* screen of the MDO Query , you will see a property as Native Mode. On clicking on the check box, it will allow the user to execute using Native SQL statements.

5. **What is the difference in Joined MDOs and MDO Join Query mode when they are doing the same by performing joins on Persistent MDOs?**

   There is a difference between both the Joined MDOs and the MDO Join Query where:

   - *Joined MDO* supports Design Time Joins and you can perform it on Persistent MDOs only.
   - *MDO Join Query* supports Run Time joins and you can perform them on both Persistent MDOs and Joined MDOs.

**6. I created a Persistent MDO with a data Provider and created some attributes to which one of them has a default value as NULL by clicking on the 'Set to Null' button. On save I see an error dialog box which is as below:**




The error is thrown since the attribute that has been set to null and in order for the validation to pass so that even null default values can be allowed, you can select the check box '*Allow Null Default Values*' .