CHAPTER 7

New Features of SAP MII

With the advancement in the technology, SAP MII has evolved with new patches and upgraded versions. It started with SAP MII version 11.5 and the latest version is 15.1.

This chapter explains the new features available in SAP MII. The following features of SAP MII are covered in this chapter:

- MDO
- Visualization services
- UI5 and SSCE
- KPI and alerts
- PIC
- PCo
- Session handling in queries and JRA
- New action blocks from SAP 15.X

What Are MDOs?

MDOs are the manufacturing data objects that can store data in the SAP MII backend application DB, which is received from different data sources and configured and integrated with SAP MII. It is used to define the data object model's structure and to persist the data. This feature was added in the 12.2 version of SAP MII and was further enhanced in later versions.

When you need to refer to a set of data that's not being updated too frequently, it is best to use an MDO as storage. Though SAP MII can integrate with external storage systems, many times, due to an external database server crash, connectivity issues, and security access issues, SAP MII will not be able to fetch data from an external database. Apart from that, external databases require additional cost and maintenance. They need additional expert resources to handle them. SAP has designed the MDO in such a way that if SAP MII uses the MDO, it will be stored in the backend application. MDO storage will happen in standard DB; otherwise, if SAP MII is installed on top of the HANA platform, the MDO storage will be in HANA DB. HANA in-memory processing is faster so MDO data processing will also be faster if it's stored in HANA DB. No additional cost is required to keep the data in MDO, and developers with SAP MII expertise can also handle the MDO and its design.

S. Mukherjee and S. Mukherjee (Das), *SAP MII*, DOI 10.1007/978-1-4842-2814-2_7

How MDOs Work

As the name suggests, MDOs are instances of manufacturing data objects. Developers can create MDOs from the Object tab in Workbench, just like any other object type.

As soon as an MDO object is created in the MII Workbench, a data object is created in the SAP MII database. Object creation involves creating a table with column definitions. That newly created data object points to a newly created database table specific to that MDO. Any data operation done to that MDO will in turn be stored in the corresponding database table.

SAP MII provides three types of MDOs:

- Persistent MDOs
- On-demand MDOs
- Joined MDOs

Persistent MDOs

These are like physical database tables. Internally, SAP MII creates a data table for any persistent MDO that is available in the same database where SAP MII is installed and uses the same underlying application database schema. Persistent MDOs can retain the data stored in the MDO until it is manually deleted.

On-Demand MDOs

No physical data tables are created for on-demand MDOs. There is no option to store data in the database. Data is populated on demand. When data from an on-demand MDO is queried, the Data Provider runs in the backend to provide datasets at runtime and the data exists until runtime. Data instances cannot be retrieved once the MDO is executed.

Joined MDOs

These are joins of two or more persistent MDOs to get a combined view of multiple MDO tables. Joined MDOs support INNER joins and LEFT OUTER joins between the MDOs.

Creating MDOs

The following sections describe how to create the MDO types in SAP MII.

Creating a Persistent MDO

Follow these steps to create a persistent MDO:

1. Open SAP MII Workbench.
2. Open the Object tab.

3. Right-click on the folder where the MDO object needs to be created.
4. Select New File ➤ Persistent MDO, as shown in Figure 7-1.

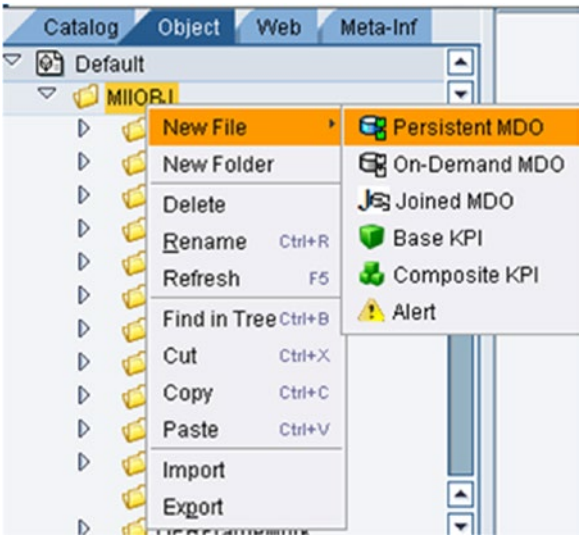


Figure 7-1. *Creating a persistent MDO*

5. The list of available screens for an MDO can be found at the bottom-left corner of the page under the Template Categories tab (see Figure 7-2). The available configurations are as follows:
    - Data Provider
    - Attributes
    - Properties
    - Lifecycle
    - Visualization
    - Security

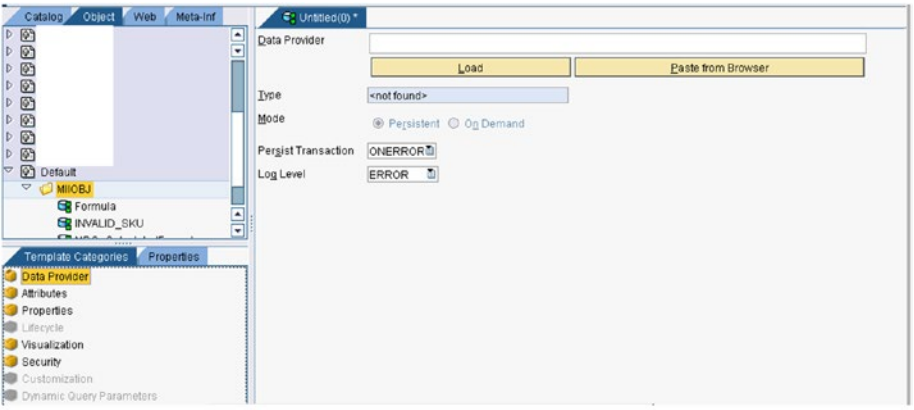


Figure 7-2. *The MDO Data Provider Screen*

Data Provider

Data Provider is the object from which data would be stored in the MDO. This configuration provides initial data to the MDO. This is an optional configuration.

1. To define the Data Provider, click on the Load button, as shown in Figure 7-3.

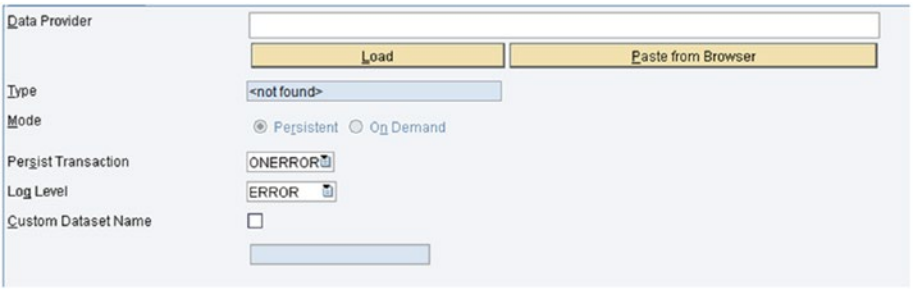


Figure 7-3. *Defining a Data Provider*

2. Select the object (BLS or Query template) from the available tree, as shown in Figure 7-4.

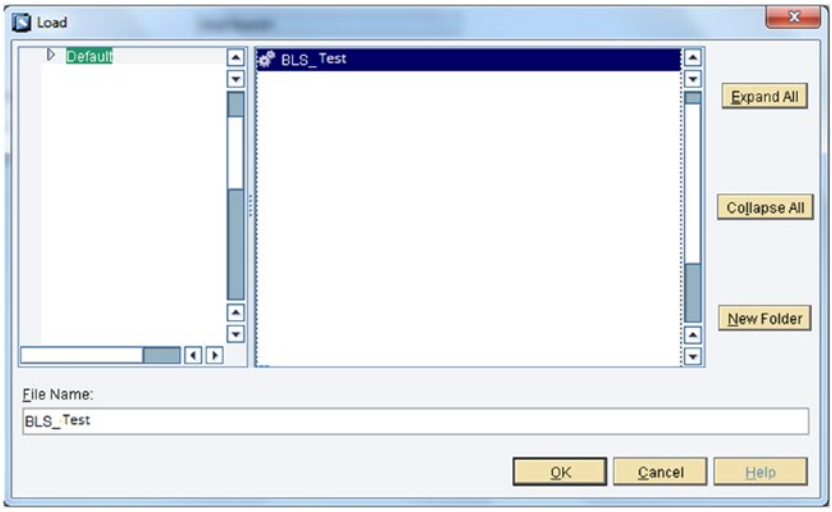


Figure 7-4. *Loading the object*

3. Click on OK to confirm the object for Data Provider and then select the Attributes configuration from the left template panel.

Attributes

This configuration defines the data structure and design for the data model. Attributes are mandatory.

1. When you select the Attributes configuration, the screen in Figure 7-5 appears.

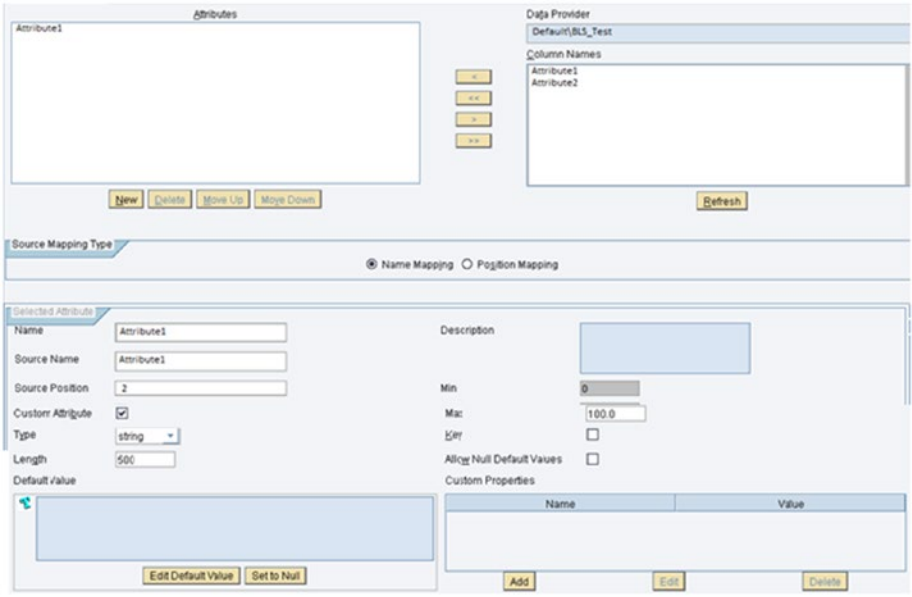


Figure 7-5. *MDO Attribute Configuration screen*

Column Names lists all columns as per the Data Provider. The required columns can be dragged and dropped under the Attribute list. If no Data Provider is mentioned initially, then it is mandatory to define the attributes by clicking the New button and assigning the properties of the attributes.

The Source Mapping Type radio buttons specify whether you want to map the attributes by name or position. You can modify the attribute values. Two different types of attributes are:

- Data attribute (directly mapped to the columns of the Data Provider)
- Custom attribute (not mapped to the columns of the Data Provider and newly created by the developer)

Source Name: In the case of the Data Attribute type, the source name links the MDO attribute name to the Data Provider attribute name and must exactly match the Data Provider attribute name.

Source Position: When the Data Provider is used then the source position can be provided and the attribute takes in the output as returned in the dataset by the Data Provider.

Custom Attribute: This feature is bound to the attribute whenever any new attribute is created without reference to a Data Provider (source name and position).

Key: If this checkbox is selected, the attribute behaves like a primary key of the MDO table structure.

Allow Null default Values: If this is checked then the column will allow a default null value into the MDO table; otherwise, the column has to be inserted with a value.

A default value can be defined for that column, so in case no value is passed to MDO table, it can take the default value defined against the attribute column.

Lifecycle

Lifecycle can be used to define the duration the data will persist in the MDO. This is applicable only to persistent MDOs. This is visible only when the Data Provider is selected. See Figure 7-6.

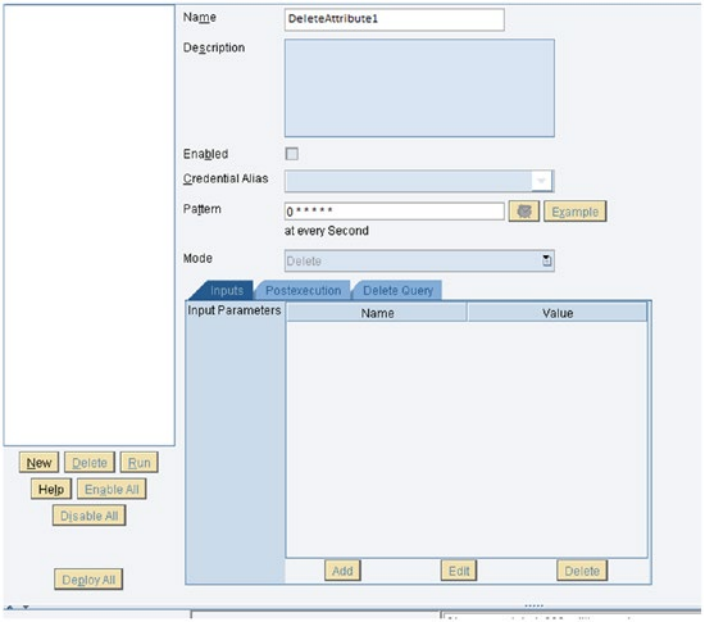


Figure 7-6. *MDO Lifecycle configuration*

Any Task can be scheduled based on the pattern defined. You need to provide a credential alias to execute it from the scheduler. The following types of modes are available:

- *Replace*: When the MDO Data Provider is executed, the persisted data is deleted and replaced with the resulting dataset.
- *Update*: When the MDO Data Provider is executed, the persisted data is updated with the resulting dataset, which is based on the key attributes for the MDO.
- *Delete*: When the MDO Data Provider is executed, the persisted data stored in the MDO will be completely deleted.
- *Inputs*: The Data Provider Transactional or Query Input parameters can be provided by default values for runtime execution.
- *Post-execution*: This defines any transaction found to be necessary as part of the dependent logic in the process, after the MDO is executed with Replace, Update, Delete mode.
- *Delete Query*: This defines the Query template to be executed to delete the MDO data. This tab is enabled only if the Delete mode is selected.

Security

This configuration defines the Reader and Writer role sections for the MDO object to be executed.

1. The Reader and Writer roles can be added as available in NetWeaver from the available roles drop-down, as shown in Figure 7-7.

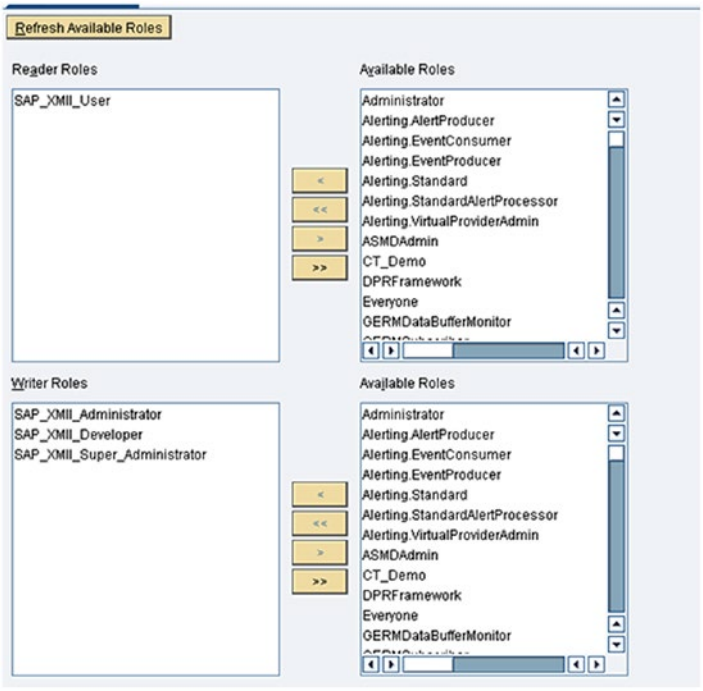


Figure 7-7. *The MDO Security configuration*

The persistent MDO is created and executed now.

Creating an On-Demand MDO

On-demand mode MDOs refer to data that is retrieved from the defined Data Provider when a query request is issued to the object. This data can be all of the object data or a filtered subset of the Data Provider content, depending on your reporting needs. It does not persist the data, so no INSERT/UPDATE/DELETE MDO queries can be used here. Step-wise creation is the same as with a persistent MDO.

Creating a Joined MDO

Follow these steps to join two or more MDOs:

1. Open the SAP MII Workbench.
2. Open the Object tab.
3. Right-click on the folder where the MDO object needs to be created.
4. Select New File ➤ Joined MDO, as shown in Figure 7-8.

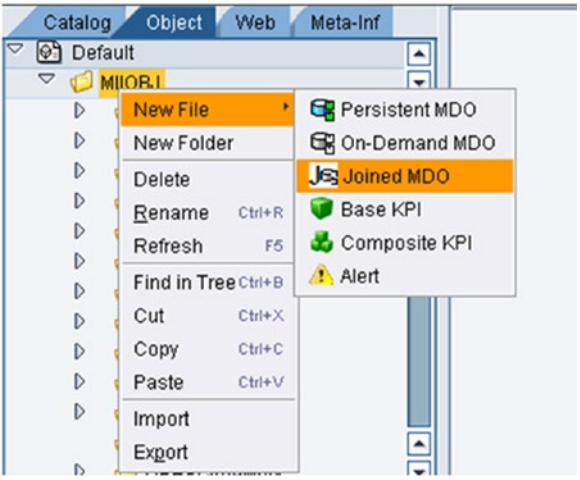


Figure 7-8. *Joined MDO creation*

5. When you select the joined MDO, the screen in Figure 7-9 appears.

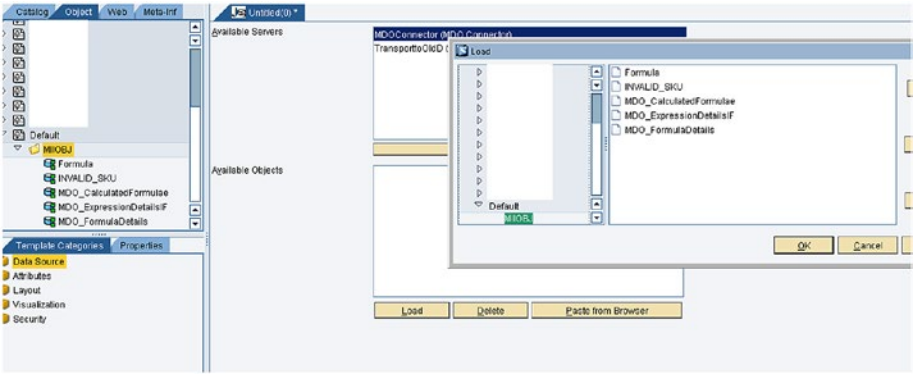


Figure 7-9. *The joined MDO object selection*

The following configurations are possible:

- Data Source
- Attributes
- Layout
- Visualization
- Security

Data Source

Data Source provides the configurations for selecting server and MDO objects to apply to the join. The available servers option lists the MDO connectors configured on the MII server, among which only one MDO connector needs to be selected.

The available objects option enables you to browse through all the project's MDOs available on the server. You can load multiple MDOs from multiple projects from the available list.

Attributes

You can select attributes from the respective available MDO objects. At least one condition needs to be defined for the INNER or LEFT OUTER INNER join (see Figure 7-10). Further, a WHERE clause could be specified with the help of filter expressions. The following options are available:

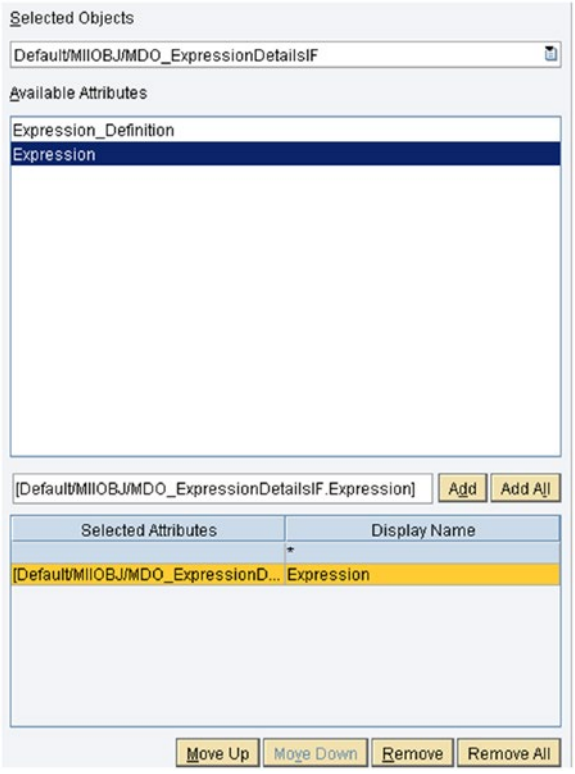


Figure 7-10. *The joined MDO attribute configuration screen*

- *Selected Objects*: This lists all the MDOs selected in the previous step for JOIN operation.
- *Available Attributes*: This lists the attributes for the selected MDOs from Selected Objects list.
- *Selected Attributes*: Whatever attributes are selected and added for the JOIN operation will be listed under this section table.
- *Display Name*: You can change the JOIN output name of the column as desired.

Figure 7-11 shows how join types can be selected.

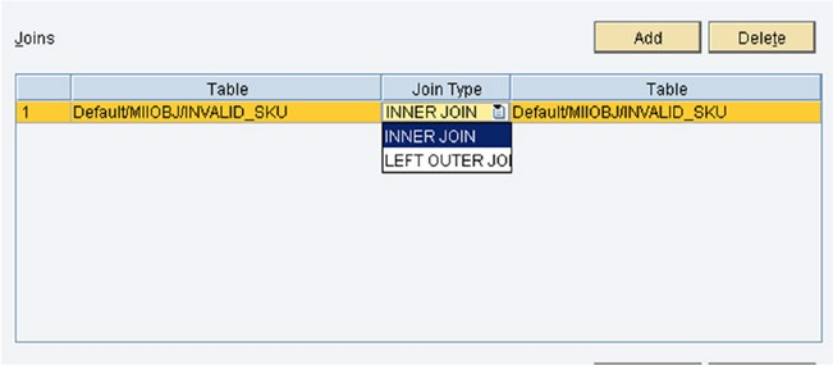


Figure 7-11. *Available joins in a joined MDO*

■ **Note** An INNER join is the same as a JOIN. They are also referred to as EQUIJOINs. The SQL INNER join returns the records where `table1` and `table2` intersect an INNER join. See Figure 7-12.

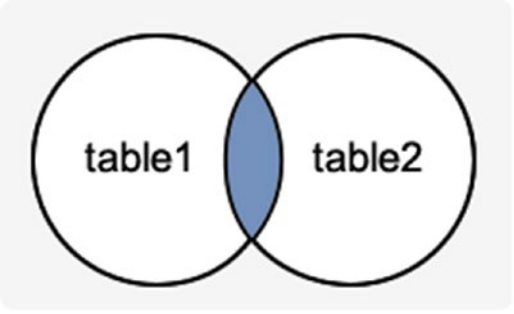


Figure 7-12. *Example of an INNER join*

LEFT OUTER join: The LEFT OUTER join returns all rows from the left-side table specified in the ON condition and only those rows from the other table where the joined fields are equal (where the join condition is met). See Figure 7-13.

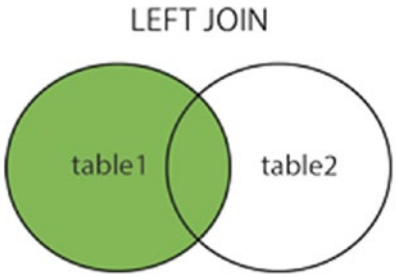


Figure 7-13. *Example of a LEFT OUTER join*

1. At least one condition needs to be defined between the MDOs to apply the join to the available operators, as shown in Figure 7-14.

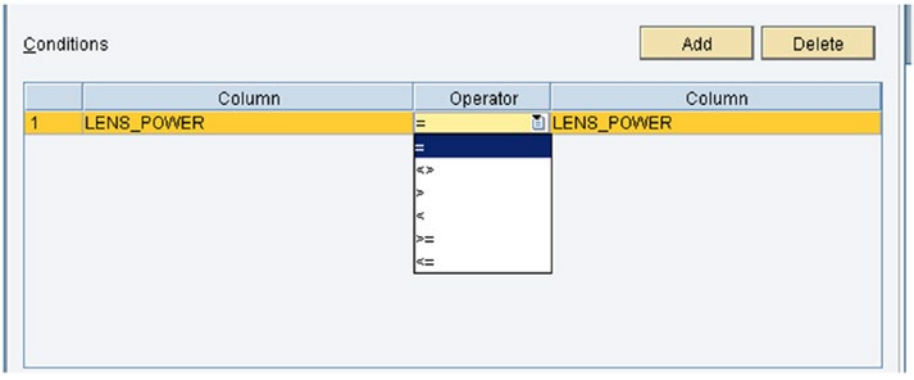


Figure 7-14. *Condition available in a joined MDO*

2. Further, you can filter the attributes selected for the join output, as shown in Figure 7-15.

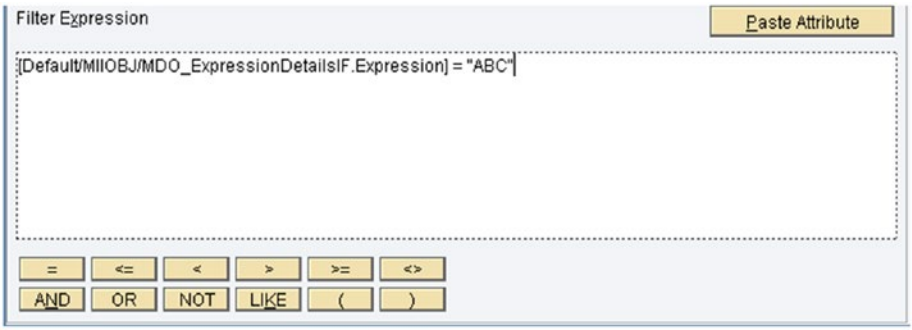


Figure 7-15. *Filter expression in a joined MDO*

Layout

Displays the graphical view of the joins between the MDO objects. This screen reflects any changes made on the Attribute tab.

Security

The same as for persistent and on-demand MDOs.

Limitations of MDOs

Although MDOs are free, they have some limitations. Following are some of the limitations of MDOs:

- They are not recommended for huge data processing.
- They are not recommended when there are frequent hits of high data volume.
- They are not recommended for querying a high volume of data at a single execution call.

Details of the MDO tables created in MII are stored in the backend standard tables using a unique identification number (such as MDO3567) and not with the name the user provided while creating the MDO. That makes it very difficult to identify specific MDOs from the backend MII standard table through Illuminator services. SAP will likely change the design of MDOs in the next release so that it will be easier to call MDOs when fetching the different details dynamically.

Visualization Services

Data visualization is the practice of visual communication (descriptive statistics) that involves study and visual representation of data. Data visualization services visualize data interactively. They begin with data analysis and end with final designed and developed product. A final product can be interactive reports, dashboards, interactive presentation apps (mobile or web), etc.

Like many other platforms, SAP MII let users create dashboard objects and web pages that can be used in various kinds of applications. SAP MII visualization services are divided into two parts:

- Dynamic page generators
- Animated objects

Dynamic Page Generator

This creates an SAP MII based HTML page content or chart servlet URL. This can be accomplished by selecting the appropriate display template and query template and various other properties related to dimensions and other configurations.

Follow these steps to create display templates:

1. On the SAP MII administration menu, choose Content Development ➤ Workbench.

The SAP MII Workbench screen appears. Do one of the following:

2. Choose File ➤ New to display the template types you can create.

There are display templates that are applet based and there are some newly added ones that are UI5 based. See Figure 7-16.

The different display templates available are as follows:

- iGrid
- iTicker
- iChart
- iSPC Chart
- iBrowser
- iCommand
- iCalendar
- i5Chart
- i5Grid
- i5SPC Chart
- i5ValueHelp

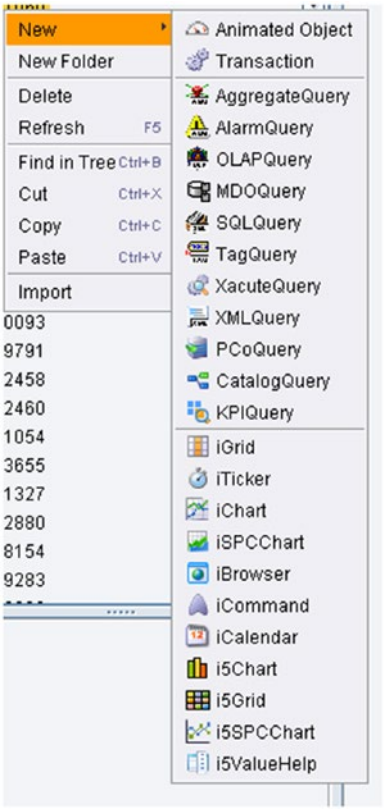


Figure 7-16. *List of display templates in SAP MII*

Apart from these display templates, the Animated Objects option is also available and SVG images can be altered using it (Animated Objects are covered later in this chapter). The following sections cover some frequently used display templates.

iGrid

Data can be viewed using the Grid display template using SAP MII connectors. You can define color context-sensitive highlighting to accentuate data elements based on data values or ranges. A grid can be used to present a single or multiple selection listbox to the users. Here are the iGrid properties that can be configured from the SAP MII Workbench (see Figure 7-17):

- Grid Area
- Layout
- Header
- UI Behavior

- Row Heading
- Color Context
- Context Menu Behaviors
- Refresh Page
- Security

Figure 7-17. *iGrid configuration*

iTicker

Enables a scrolling stock ticker that can display data returned from any SAP MII connector. The iTicker uses the applet height and width tags to automatically size the applet window. Ticker text is automatically centered vertically. Data values can be highlighted in different colors in the ticker.

Configurable iTicker properties include the following (see Figure 7-18):

- Ticker Area
- Data Mapping
- Color Context
- Refresh Page
- Security

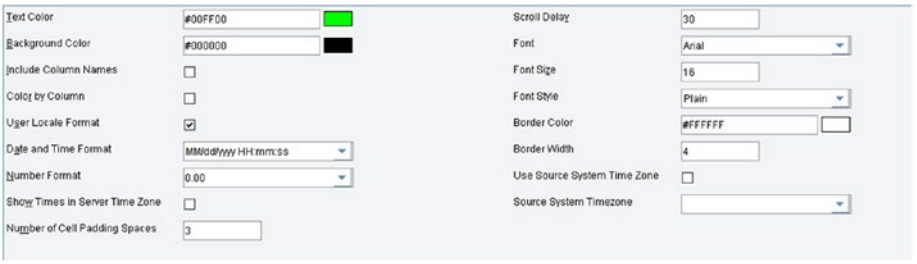


Figure 7-18. *iTicker configuration*

iChart

Processes and displays datasets with up to 32 tags and thousands of observations per tag in a variety of chart types. SAP MII uses the iChart applet, which supports the `SelectionEvent` applet parameter and allows JavaScript functions to be linked to items in the chart legend.

iChart properties can be configured on the following screens of the SAP MII Workbench (see Figure 7-19):

- Chart Area
- Title
- Data Mapping
- Legend
- X-Axis
- Y-Axis
- Server Scaling
- Data Series Details
- Context Menu Behaviors
- Refresh Page
- Security

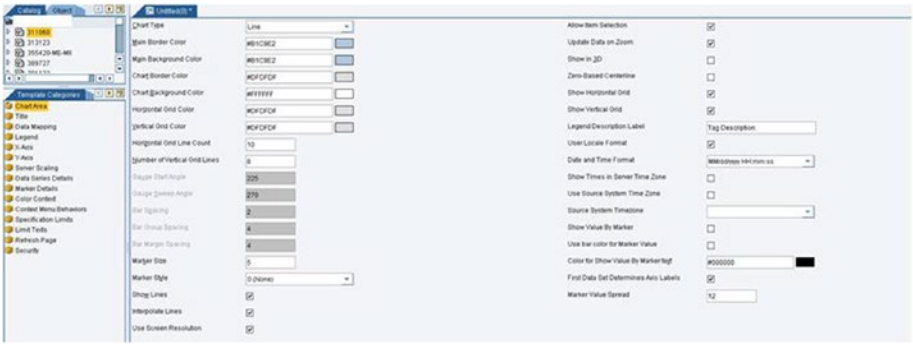


Figure 7-19. *iChart configuration*

iSPC Chart

Processes and displays datasets with thousands of observations per tag in a variety of statistical process control (SPC) chart formats. Each SPC chart includes the following (see Figure 7-20):

- A set of summary statistics appropriate for the analysis type
- Raw data views and full statistical calculation result views
- SPC rule violations with drill-down capability on a point-by-point basis
- Additional attribute information for each point
- Collaborative point documentation capabilities

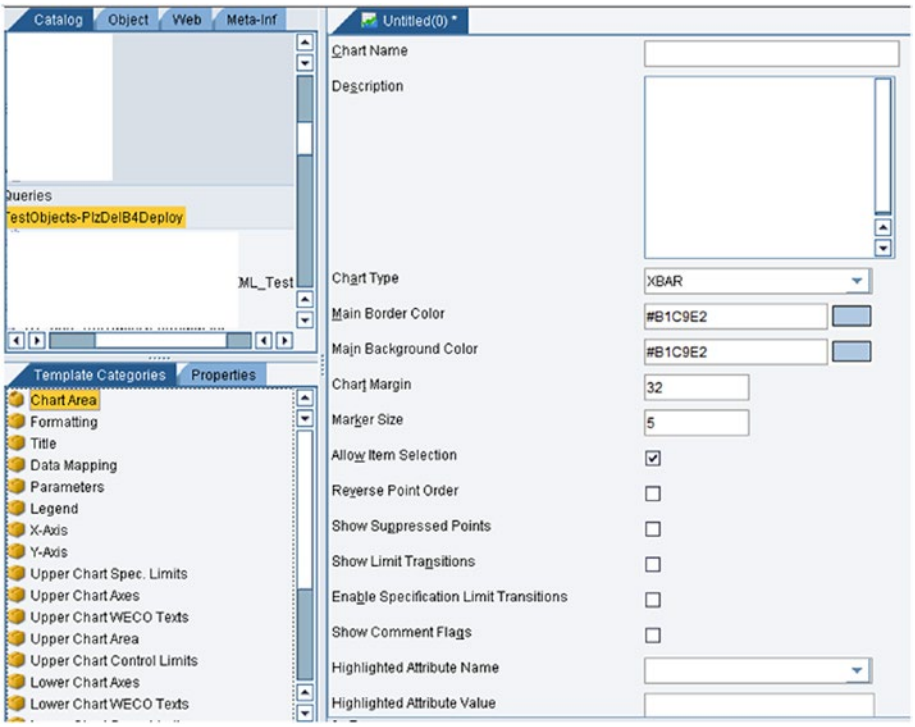


Figure 7-20. *iSPC chart configuration*

Figure 7-20 shows different views or multi-dimensions of the data that can be displayed while interacting with the SPC charts of the application. When the user clicks on a data point in the SPC chart, the screens are updated to show information specific to that point. If additional data of any dimension is returned from the data query linked to the SPC chart, that can be displayed as added context to the view.

Here are the iSPC chart configurable properties in SAP MII Workbench:

- Chart Area
- Formatting
- Title
- Data Mapping
- Parameters
- Legend
- X-Axis
- Y-Axis
- Upper Chart Spec. Limits
- Upper Chart Axes
- Upper Chart WECO Texts
- Upper Chart Area
- Upper Chart Control Limits
- Lower Chart Axes
- Lower Chart WECO Texts
- Lower Chart Spec. Limits
- Lower Chart Area
- Lower Chart Control Limits
- Alarms
- Limit Texts
- Context Menu Behaviors
- Context Menu Security
- Refresh Page
- Security

iBrowser

Using iBrowser, you can map data from a range of sources to the SAP MII tree, list, or drop-down list views, which provide selection, drill-down, and other functions.

When a query is created to map to a browser, the following applies:

- When a single column is returned from the query, the browser is configured as a listbox or a drop-down listbox.

- When two columns are returned and the Data Link Mode is selected, the second column shows the data link value for the item and the column is hidden. The browser is configured as a listbox or a drop-down listbox.
- When two columns are returned and the Data Link Mode checkbox is not selected, the browser is configured as a tree. The query must return the name of the item in the first column and the name of its parent item in the second column.
- When three or more columns are returned, the browser is configured as a tree. The query must return the name of the item in the first column and the name of its parent item in the second column.
- When four columns are returned and the Data Link Mode checkbox is selected, the system assumes the first column includes an item, the second column includes the parent node, the third column includes the data link value for the item, and the fourth column includes the data link value for the parent node.
- When the browser is configured as a tree, a selected node can be deselected by pressing Ctrl and clicking the node or by pressing the spacebar when the tree has focus.

Here are the configurable iBrowser properties in the SAP MII Workbench (see Figure 7-21):

- General
- Data Mapping
- Security

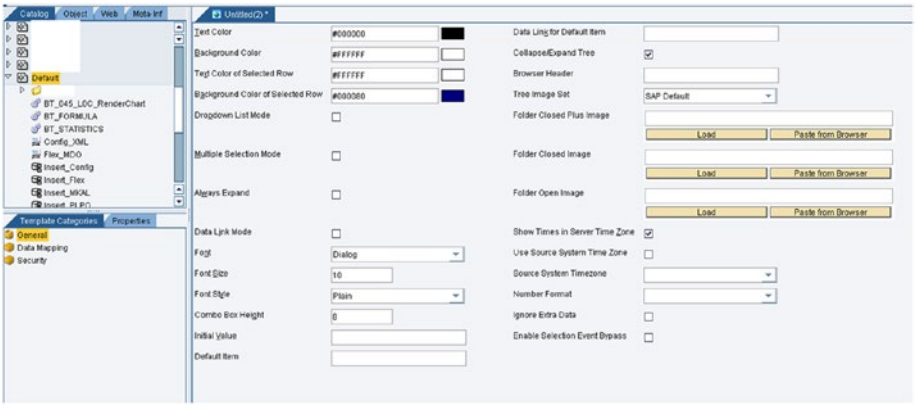


Figure 7-21. *iBrowser configuration*

iCommand

iCommand allows the execution of SQL command queries without performing an HTTP POST. Because it can hold the dataset, a data query template can be loaded into a hidden object on the screen and that data can be used in DHTML, HTML tables, other applets, etc. iCommand does not necessarily require display elements to be defined and the applet does not require any pre-execution of the template query request. That makes it more efficient than a grid.

Electronic signature and audit logging is supported by SAP MII through the iCommand applet. This functionality enables end users to implement 21 CFR Part 11-compliant solutions. There are special settings in an iCommand applet display template for these functions. See Figure 7-22.

The following configurations are available:

- General
- Security

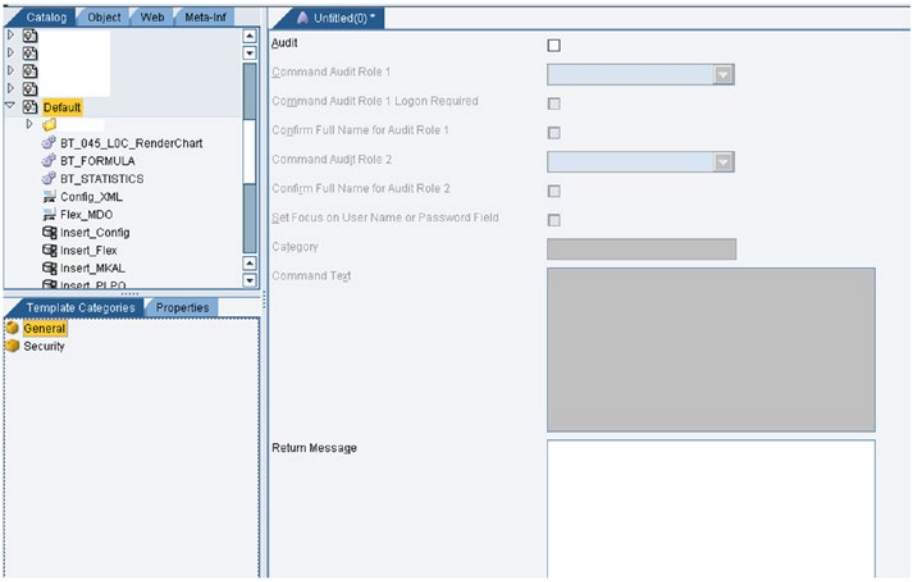


Figure 7-22. *iCommand configuration*

iCalendar

A calendar on an applet can be included and linked to other applets or web page form elements. SAP MII supports the Gregorian calendar only. It is recommended to set the applet dimensions to 200 pixels wide by 188 pixels high. See Figure 7-23.

The following configuration options are available:

- General
- Security

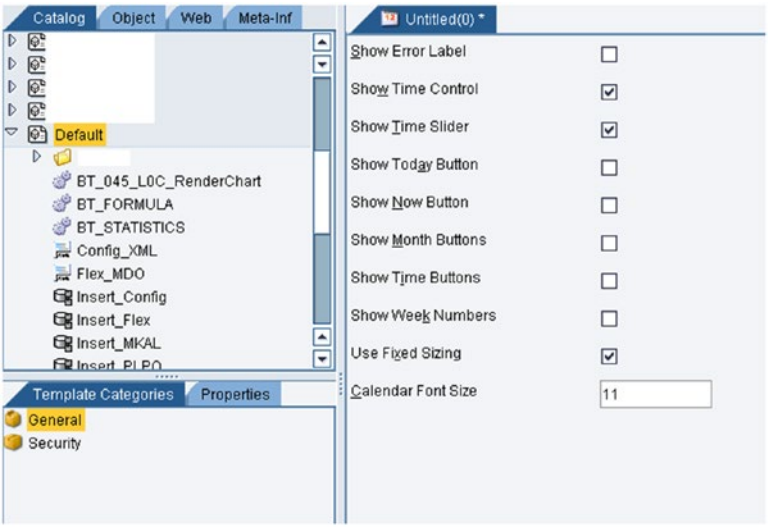


Figure 7-23. *iCalendar configuration*

i5Grid

In the latest versions of SAP MII (from MII 14.0 SP04), some UI5 based display templates have been added along with applet based display templates web page or browser-based scripting. JavaScript can be used to dynamically control the behavior and appearance of an i5Grid or similar display templates, including methods and events. These are basically grids where data from queries are being loaded. They have faster execution times than applets. See Figure 7-24.

Figure 7-24. *i5Grid creation*

Figure 7-25 shows the example of an i5Grid marked with all the available properties.

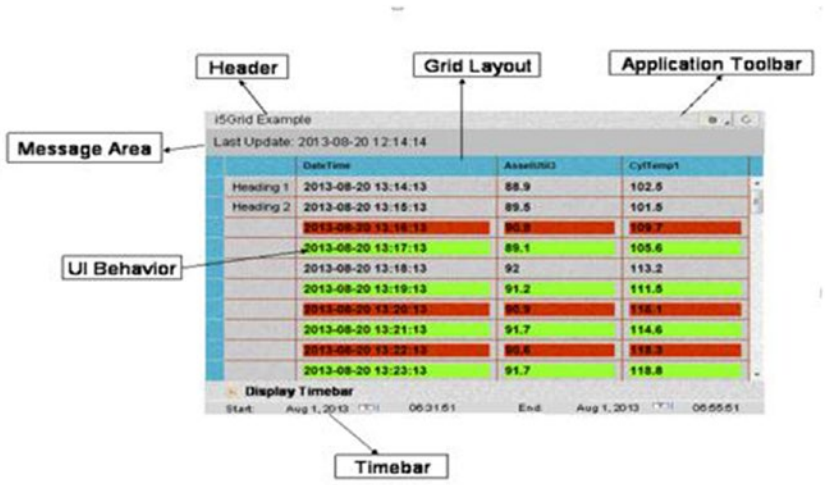


Figure 7-25. *Graphical representation of an i5Grid*

Here is a list of the configurations available in i5Grid. Grid area is one of main configurations to populate the grid. See Figure 7-26.

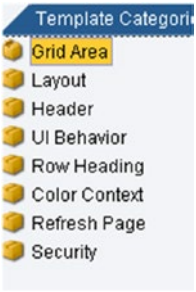


Figure 7-26. *i5Grid configuration options*

- *i5Grid Area*: Grid Area configuration configures the general properties of the display template. In Grid Area, grid type can be specified. See Figure 7-27.

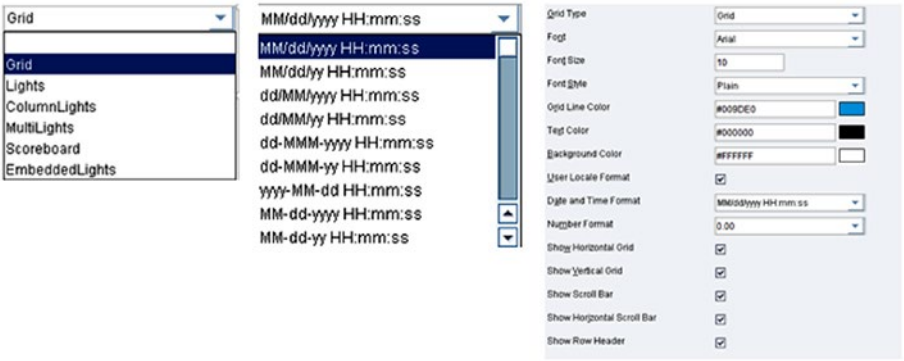


Figure 7-27. *Various configurations of i5Grid*

- *i5Grid Layout*: Layout configuration configures the data source and the column properties of the i5Grid. See Figure 7-28.

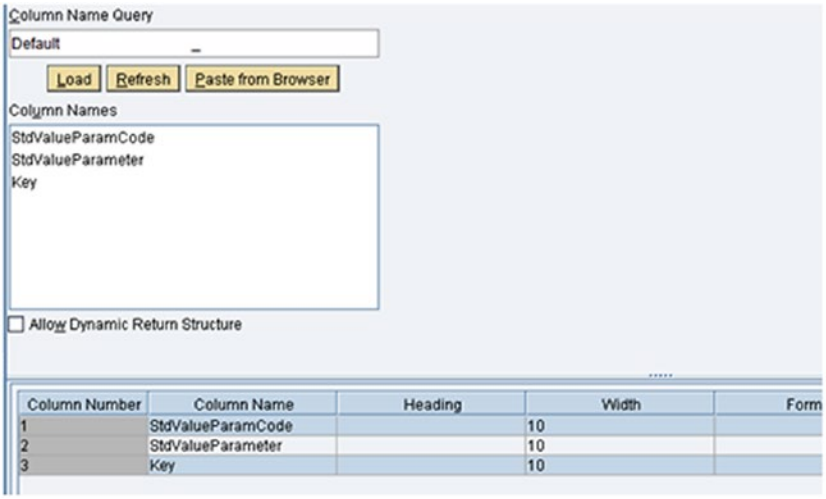


Figure 7-28. *i5Grid layout configuration*

- *i5Grid Header*: This configures the header property. See Figure 7-29.

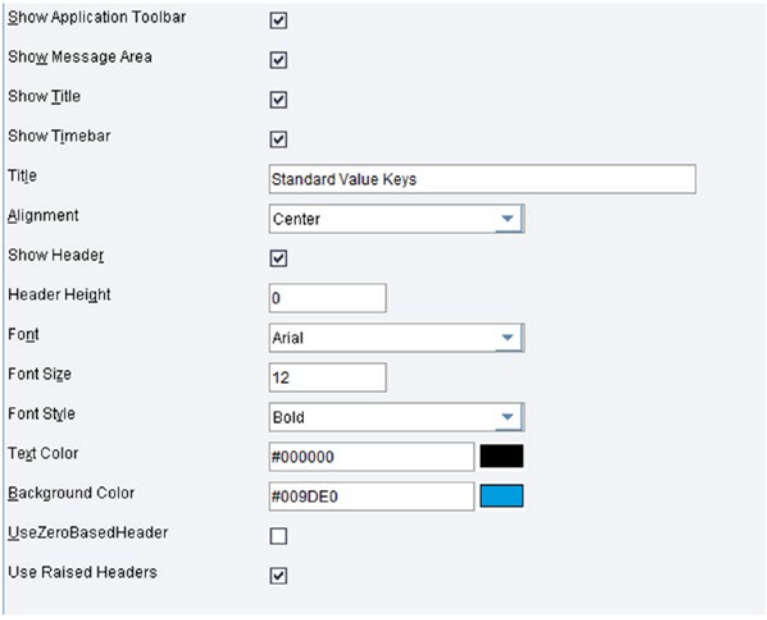


Figure 7-29. *i5Grid header configuration*

- *i5Grid UI Behavior*: User interface behavior of the i5Grid can be defined. See Figure 7-30.

StdValueParamCode	StdValueParameter	Key
SAP_01	Setup	A001
SAP_02	Machine	A002
SAP_03	Labor	A003

Figure 7-30. *i5Grid UI behavior sample*

The UI behavior can be configured in i5Grid display, as shown in Figure 7-31.

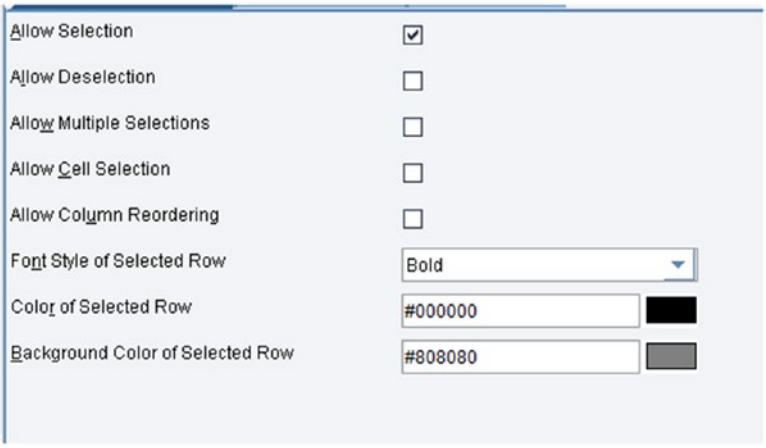


Figure 7-31. *i5Grid UI behavior configuration*

- *i5Grid Row Heading*: Row headings of SAP MII grids can be set. See Figure 7-32.

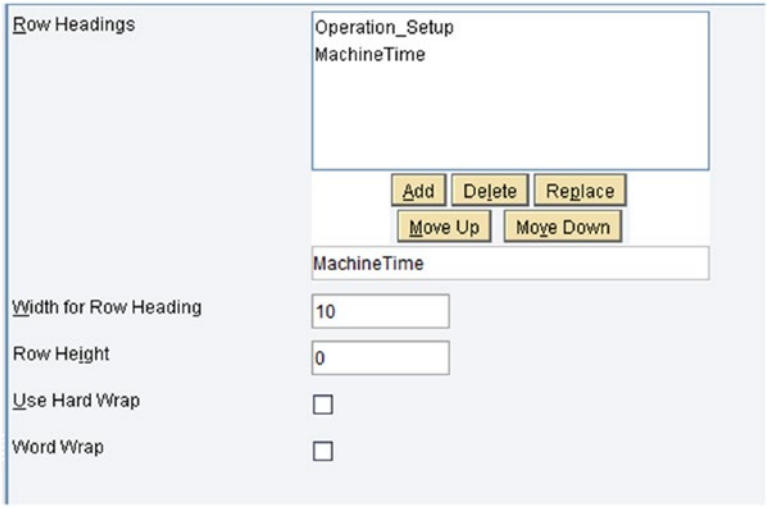


Figure 7-32. *i5Grid row heading configuration*

Once the row heading is set, the grid will look like Figure 7-33.

		StdValueParamCode	StdValueParameter	Key
	Operation_Setup	SAP_01	Setup	A001
	MachineTime	SAP_02	Machine	A002
		SAP_03	Labor	A003

Figure 7-33. *i5Grid row heading sample*

- *i5Grid Color Context*: The color context of the grid can be set. See Figure 7-34.

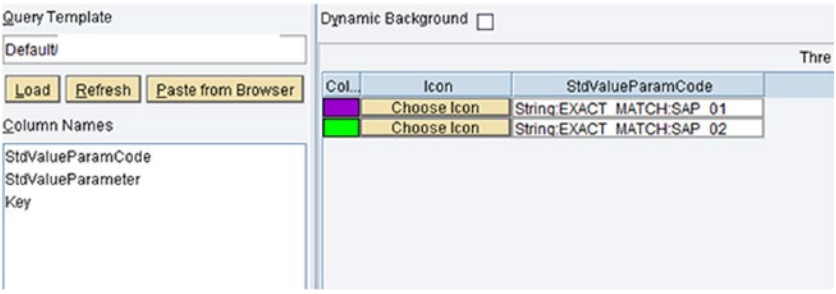


Figure 7-34. *i5Grid Color Context*

Once the Color context is configured as shown in Figure 7-34, the text of the row is displayed in the color against the conditions defined. See Figure 7-35.

StdValueParamCode	StdValueParameter	Key
SAP_01	Setup	A001
SAP_02	Machine	A002

Figure 7-35. *i5Grid color context sample*

- *i5Grid Refresh Page*: Used to configure the Refresh option, as shown in Figure 7-36.

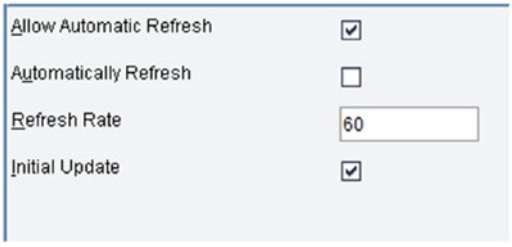


Figure 7-36. *i5Grid refresh page configuration*

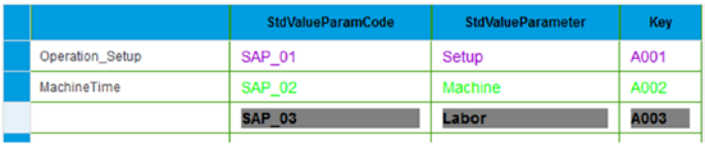
i5Grid Output

Web page or browser-based scripting can be used to dynamically control the behavior and appearance of an i5Chart in SAP Manufacturing Integration and Intelligence (SAP MII). Scripting can to do the following:

- Change i5Chart properties
- Call i5Chart methods
- Use i5Chart events, such as user selections, to link i5Chart for drill-down or data correlation applications

Figure 7-37 shows the final output sample. The code to include the i5Chart in the HTML page is as follows:

```
var i5Grid = new com.sap.xmii.grid.init.i5Grid("Default/Saptaparna/i5Grid_
Exercise", "Default/Saptaparna/MDO_ReadSVKs");
i5Grid.setGridWidth("1240px");
i5Grid.setGridHeight("500px"); i5Chart.draw(“GridDiv");
```



	StdValueParamCode	StdValueParameter	Key
Operation_Setup	SAP_01	Setup	A001
MachineTime	SAP_02	Machine	A002
	SAP_03	Labor	A003

Figure 7-37. *i5Grid final output sample with all features*

i5Grid Features

i5Grid has the following features:

- *Application Toolbar*: Located at the top-right corner of each grid and contains some icons by default.
  - *Data*: Data can be exported from the grid in three formats: HTML, CSV, and XML.
  - *Current Value*: Displays current values for all the tags available in the query. The values are displayed in HTML format in a new window for queries, which supports the Tag, PCo, and Catalog modes.
  - *Statistics*: Displays the statistical values for all the tags available in the query. The values are displayed in HTML format in a new window.
  - *Refresh Rate*: Frequency in seconds at which the grid is refreshed.
  - *Refresh Automatically*: Refreshes the grid as per the specified frequency.
  - *Print*: Prints the grid preview.
  - *Help*: Provides SAP MII Standard application help documents of the version being used.
- *Timebar*: Used to configure the hide/unhide feature of the timebar.

  The grid area displays a timebar for all query-based data. The grids are refreshed accordingly based on the selected date-time ranges. The arrows on the timebar display the charts in full screen, quarter screen, or for the current data. Date Range, Start Date, and End Date can be configured in the template or using the Date/Time Picker. The main properties are Slider, Picker, Duration, and Date/Time Picker. These new features add to the versatility of the visualization.

There are various scripting methods to return information from SAP MII i5Grid, such as `getGridObject()`, `doPrint()`, `update(Boolean ReloadData)`, and many more.

i5Chart

A new MII HTML5 based display template called i5Chart was introduced in the MII 14.0 SP04 release. i5Chart uses HTML5–based SAP UI5 technology to render charts. It provides a variety of configurable charts to display and analyze data. It is one of the most important and powerful visualization components of SAP MII. i5Chart's display template supports the `SelectionEvent` parameter and allows JavaScript functions to be linked to items in the chart. A variety of i5Chart types—such as Line, Bar, GroupBar, Pie, StackedBar, Scatter, Gauge, HorizontalBar, HorizontalGroupBar, Bubble, Donut, and Custom—are supported. See Figure 7-38.

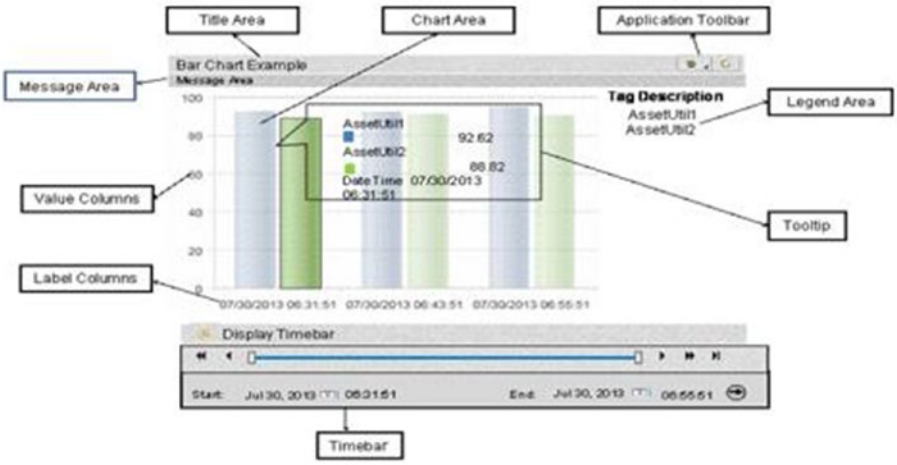


Figure 7-38. *Graphical representation of i5Chart*

- *i5Chart Chart Area*: *The* Chart Area tab configures chart type, colors, fonts, general chart parameters, and behaviors while elements are displayed in a chart. See Figure 7-39.

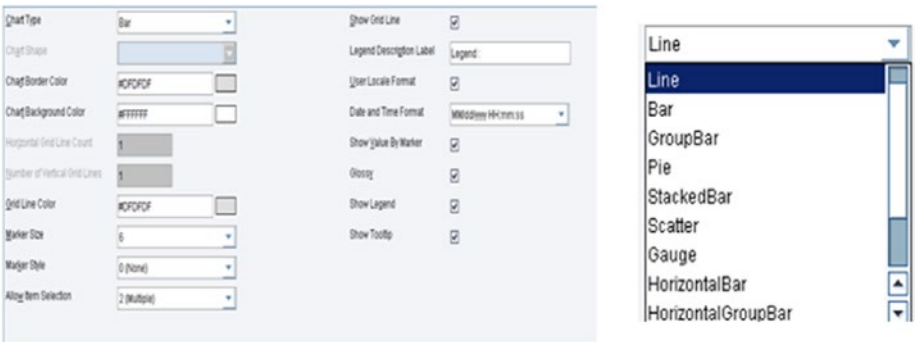


Figure 7-39. *i5Chart chart area configuration with available chart types*

- *i5Chart Title*: This sets the title, color, font, and other properties for i5Charts. See Figure 7-40.

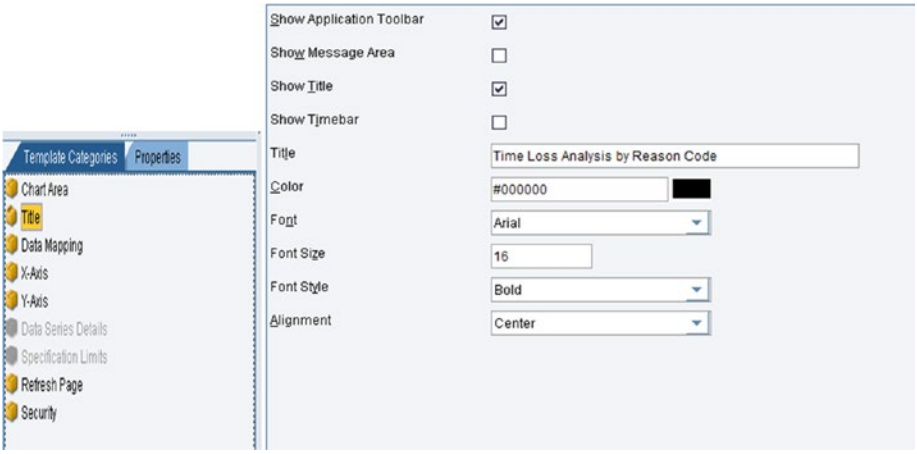


Figure 7-40. *i5Chart title configuration*

- *i5Chart Data Mapping*: Data mapping tab maps columns returned by a query to the various elements of the i5Chart object. See Figure 7-41.

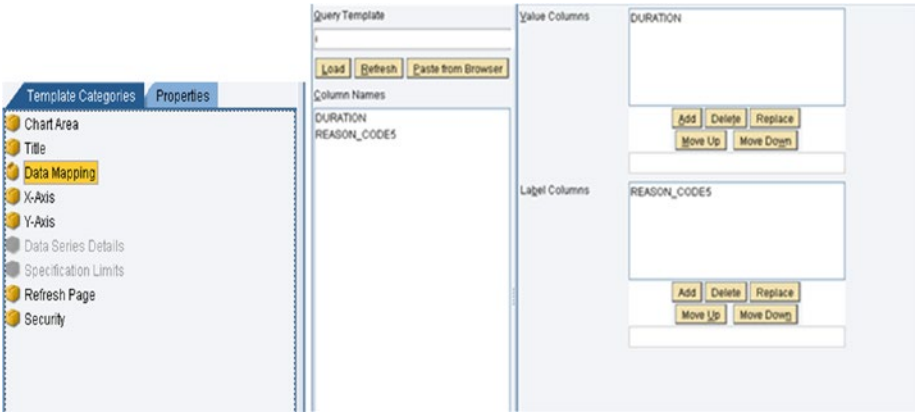


Figure 7-41. *i5Chart data mapping configuration*

- *i5Chart X-axis*: The X-axis tab configures parameters associated with the X-Axis (horizontal axis). See Figure 7-42.

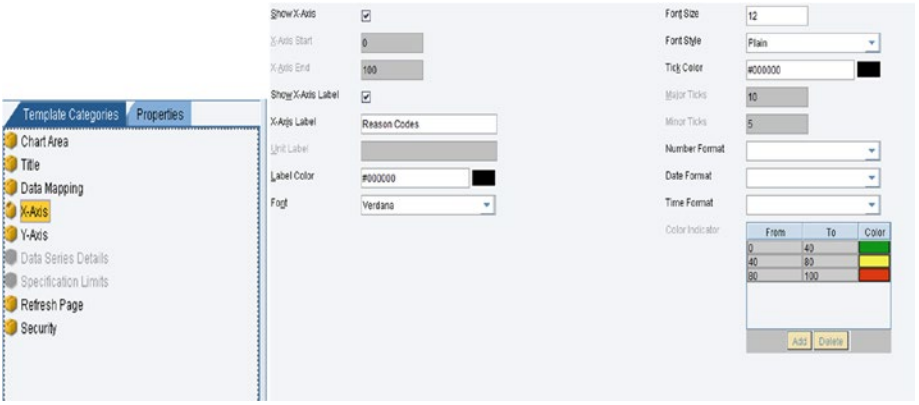


Figure 7-42. *i5Chart X-axis configuration*

- *i5Chart Y- Axis*: The Y-axis tab configures parameters associated with the Y-axis (the vertical axis). See Figure 7-43.

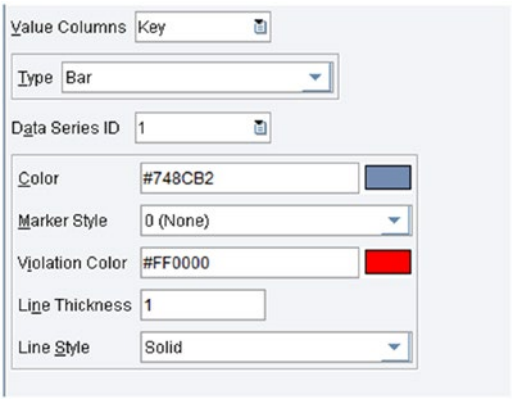


Figure 7-43. *i5Chart Y axis configuration*

- *i5Chart Data Series Details*: Data Series Details configure data series colors of the chart.
- *i5Chart Specification Limits*: The Specification Limits tab configures the specification limits for your query output by defining the upper and lower limits in the chart. See Figure 7-44.

Figure 7-44. *i5Chart specifications limit*

- *i5Chart Refresh Rate*: The Refresh Page tab sets the refresh rate and the refresh mode (automatic/manual) for the chart. See Figure 7-45.

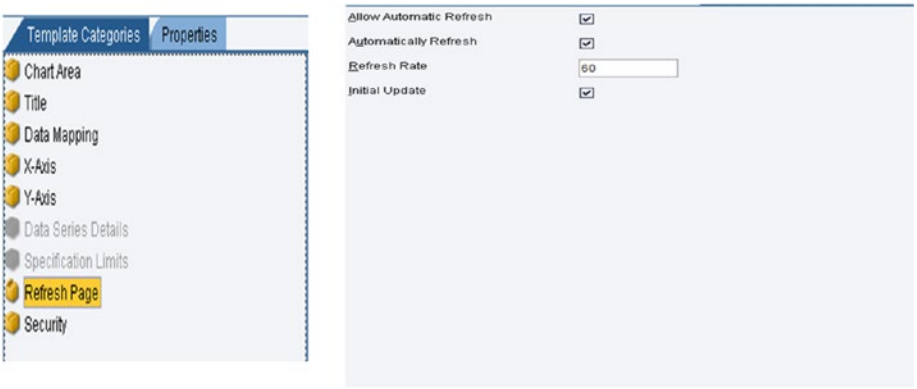


Figure 7-45. *i5Chart refresh rate configuration*

i5Chart Output

Web page or browser-based scripting, using JavaScript, can dynamically control the behavior and appearance of an i5Chart in SAP MII. Scripting can be used to:

- Change i5Chart properties
- Call i5Chart methods
- Use i5Chart events, such as user selections, to link i5Chart to drill-down or data correlation applications

Figure 7-46 shows the i5Chart final output. Here's the code to include the i5Chart in an HTML page:

```
var i5Chart = new com.sap.xmii.chart.hchart.i5Chart("Default/Chart/CHA_
I5Chart", "Default/Query/MDOQuery_GetData);
i5Chart.setChartWidth("400px");
i5Chart.setChartHeight("400px");
i5Chart.draw("chartDiv");
```

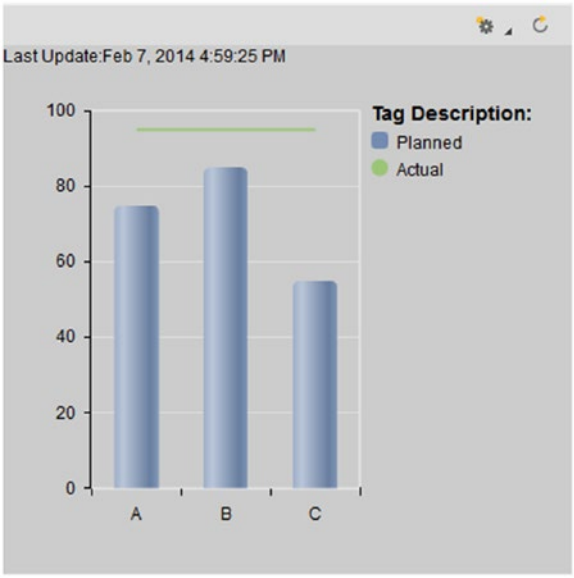


Figure 7-46. *i5Chart final output*

i5Chart Features

i5Chart has the following features:

- *Access Toolbar*: Located at the top-right corner of each chart and contains some icons by default.
  - *Data*: Data can be exported from a grid in three possible formats: HTML, CSV, and XML.
  - *Current Value*: Displays current values for all the tags available in the query. The values are displayed in HTML format in a new window for queries that support the Tag, PCo, and Catalog modes.
  - *Statistics*: Displays the statistical values for all the tags available in the query. The values are displayed in HTML format in a new window.

- *Refresh Rate*: Frequency in seconds at which the grid is refreshed.
- *Refresh Automatically*: Refreshes the grid as specified.
- *Print*: Prints the grid preview.
- *Glossy Effect*: Displays the chart with glossy effect and can be enabled or disabled in the chart area section.

- *Timebar*: Display Timebar hides/unhides the timebar. The chart area displays a timebar for all queries containing time-based data. The grids are refreshed based on the selected date-time ranges. The arrows on the timebar display the charts in full screen, quarter screen, or for the current data. Date Range, Start Date, and End Date can be configured in the template or using the Date/Time Picker. The main properties are Slider, Picker, Duration, and Date/Time Picker. These new features add to the versatility of the visualization.

There are various scripting methods to return information from the SAP MII i5Chart, including `getChartObject()`, `Refresh()`, `update(Boolean)`, and many more.

```
JavaDocs Source:
https://<server>:<port>/XMII/JSDOC/i5ChartAllClasses.html.
```

i5SPC Chart

The chart displays datasets with multiple observations per tag in a variety of Statistical Process Control (SPC) chart formats. Each SPC chart includes:

- A variety of i5SPC Chart types such as XBAR, XBAR-MR, XBAR-RANGE, XBAR-SDEV, INDIVIDUALS, INDIVIDUALS-MR, MR, EWMA, EWMA-RANGE, EWMA-SDEV, MEDIAN, MEDIAN –RANGE, HISTOGRAM, INDIVIDUALS-SHORT-RUN, INDIVIDUALS-MR-SHORT-RUN, XBAR-RANGE-SHORT-RUN, MR-SHORT-RUN, P, NP, C, U, and so on for different types of process control and statistical quality analysis.
- Data point highlighting and suppression capabilities for ad hoc analysis by users
- Auto-calculated standard statistical and quality indexes
- Visual alarms in the chart for SPC rule violation conditions and comment text addition for each data point

i5SPC chart configuration options are very similar to the iSPC chart's options. The major difference is that i5SPC is dedicated to the UI5 extension. The following template configurations are available in the i5SPC chart (see Figure 7-47):

Figure 7-47. *i5SPC chart configuration options*

When the i5SPC chart is configured, the chart is displayed as shown in Figure 7-48.

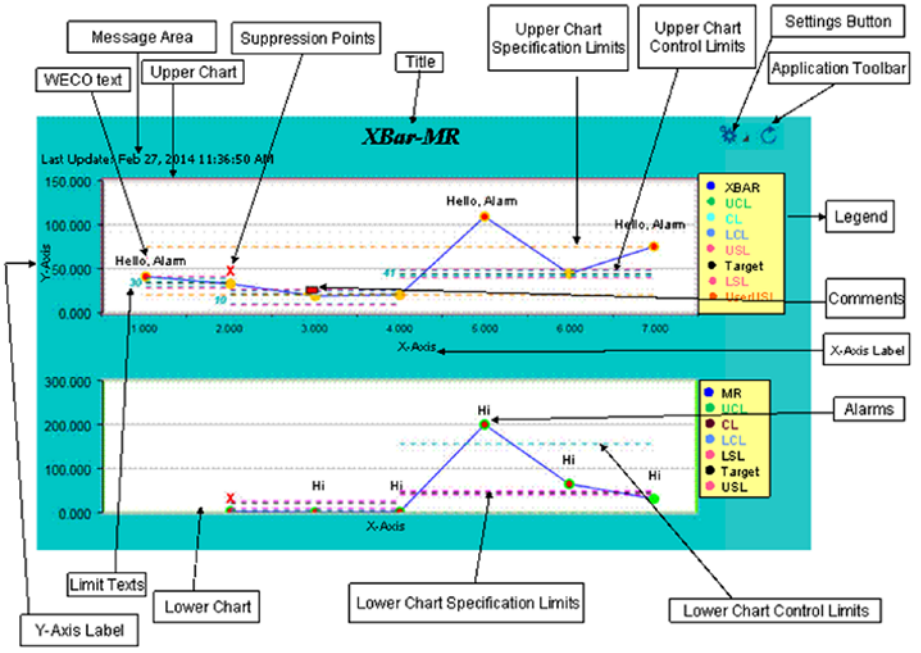


Figure 7-48. *i5SPC chart sample with all properties*

- *i5SPC chart area*: The Chart Area tab configures chart type, colors, fonts, general chart parameters, and behaviors while elements are displayed in a chart. Different possible chart types are as follows (see Figure 7-49):

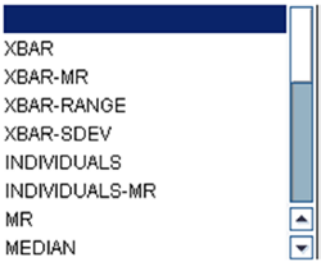


Figure 7-49. *i5SPC chart available chart types*

- *i5SPC chart formatting*: This function is available to configure the formatting properties for SAP MII SPC charts. See Figure 7-50.

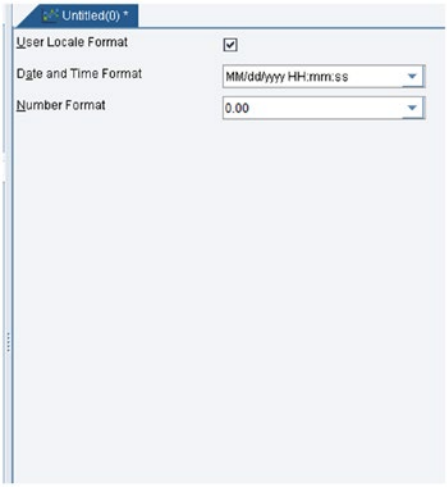


Figure 7-50. *i5spc chart formatting*

- *i5SPC chart title*: This tab sets the title, color, font, and other properties for your i5SPC charts. See Figure 7-51.

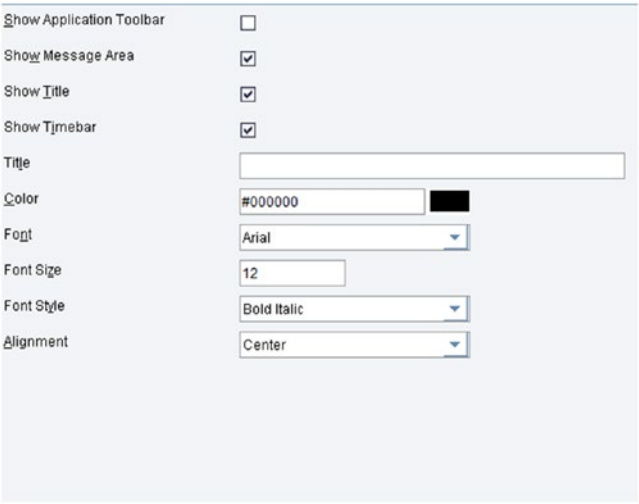


Figure 7-51. *i5SPC chart title configuration*

- *i5SPC chart Data Mapping*: The Data Mapping tab maps columns returned by a query to the various elements of the i5SPC chart object. See Figure 7-52.

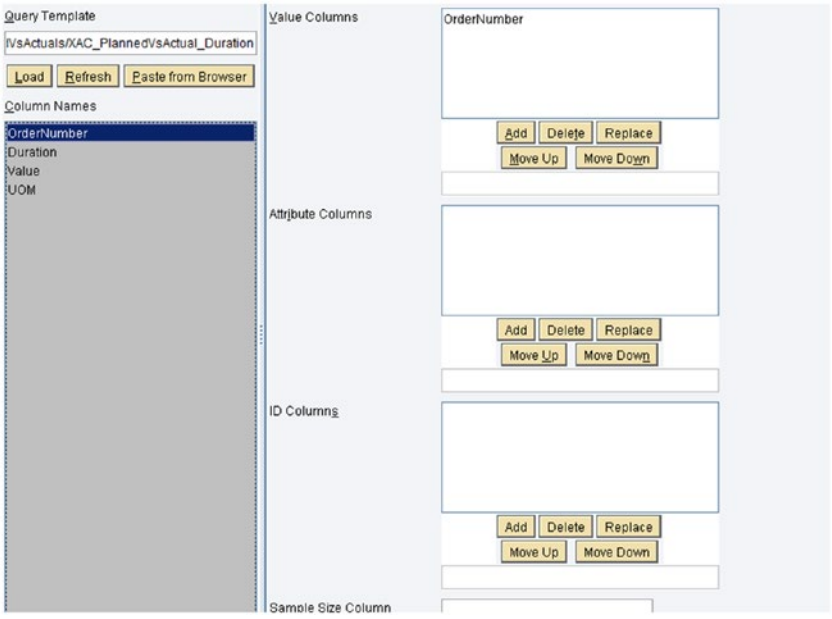


Figure 7-52. *i5SPC chart data mapping configuration*

- *i5SPC chart parameters*: The Parameter tab configures the parameters that are specific to individual chart types. See Figure 7-53.

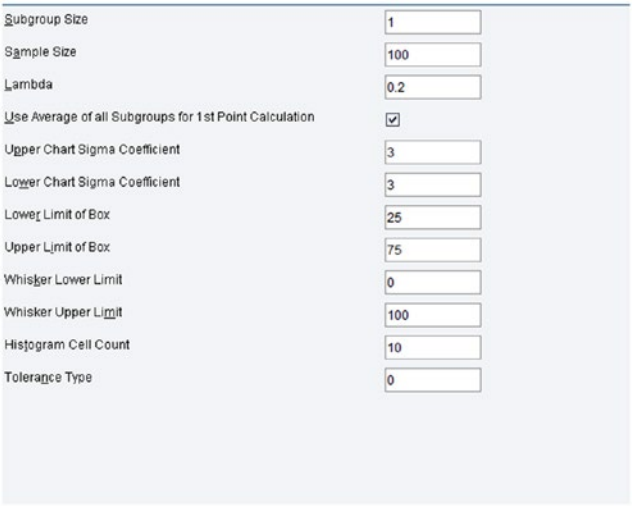


Figure 7-53. *i5SPC chart parameter configuration*

- *i5SPC chart legend*: The Legend tab configures the appearance and behavior of the chart legend that appears on the right of the SPC chart. See Figure 7-54.

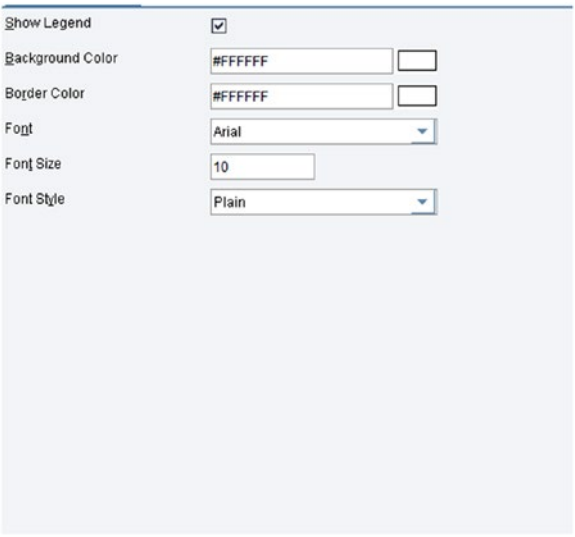


Figure 7-54. *i5SPC chart legend configuration*

- *i5SPC chart X-axis*: The X-axis tab configures parameters associated with the X-axis (the horizontal axis). See Figure 7-55.

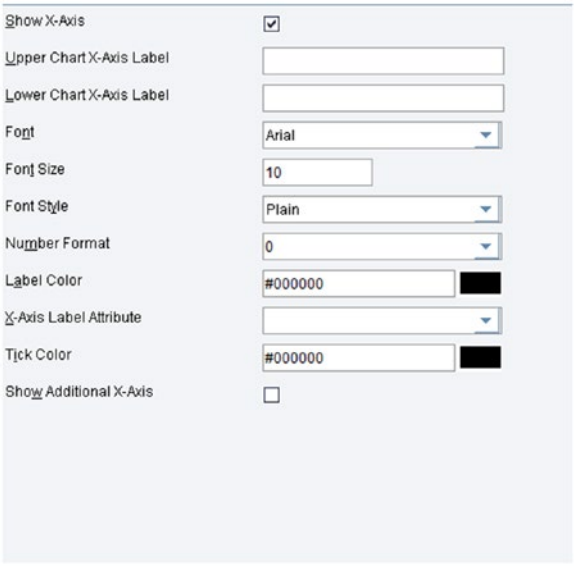


Figure 7-55. *i5SPC Chart X-Axis Configuration*

- *Y-axis tab*: Used to configure parameters associated with the y-axis (the vertical axis). See Figure 7-56.

Figure 7-56. *i5SPC chart y-axis configuration*

- *Upper chart limit*: The i5SPC chart can be configured with upper spec limits as described here (see Figure 7-57):

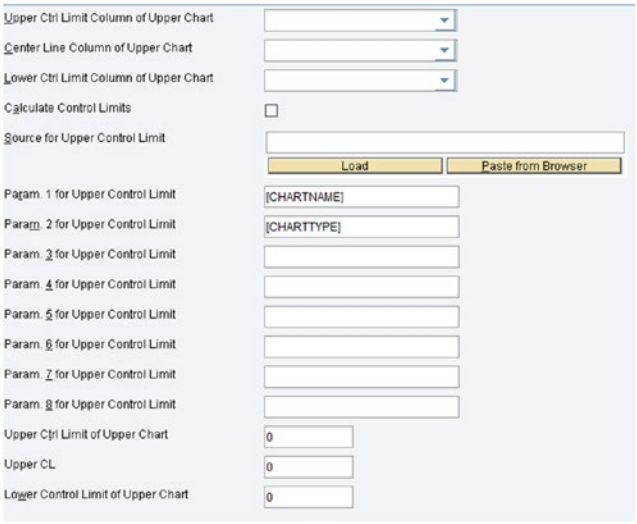


Figure 7-57. *i5SPC chart upper chart control limit configuration*

- The Upper Chart Spec Limits tab configures the specification limits of the upper chart for the statistical process control (SPC) chart.
- The Upper Chart Axes tab configures the upper chart axes of the statistical process control (SPC) chart.
- The Upper Chart Area tab configures the upper chart area of the Statistical Process Control (SPC) chart.
- The Upper Chart Control Limits configuration configures the upper chart control limits of the Statistical Process Control (SPC) chart. See Figure 7-58.

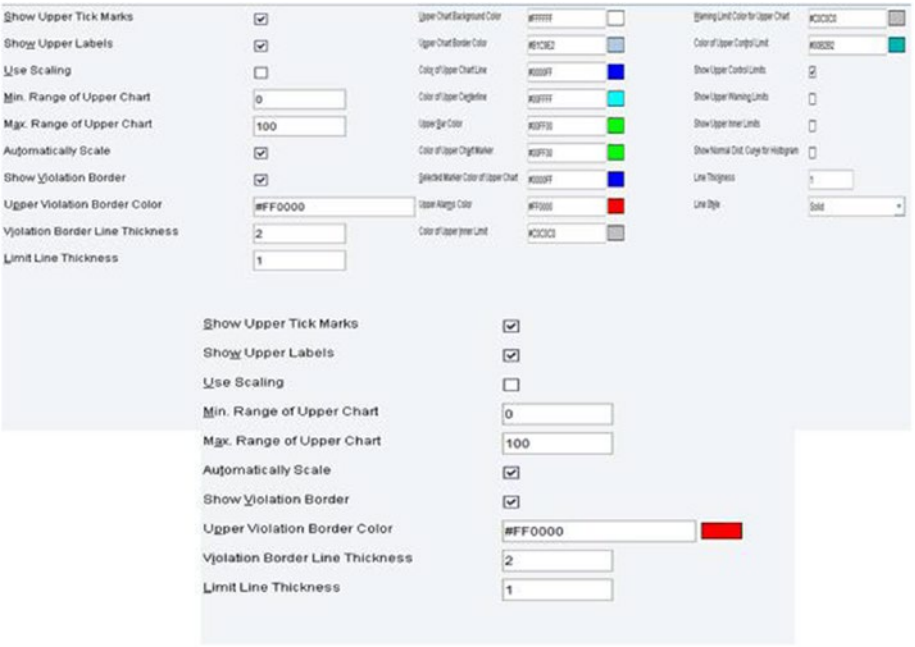


Figure 7-58. *i5SPC Chart's various control limit configuration properties*

The lower spec limits are similar, as you can see in Figure 7-59.

Figure 7-59. *i5SPC chart's lower spec limit*

- *I5SPC chart alarms*: The Alarms tab configures the alarms of an SPC chart when data values violate certain rules. See Figure 7-60.

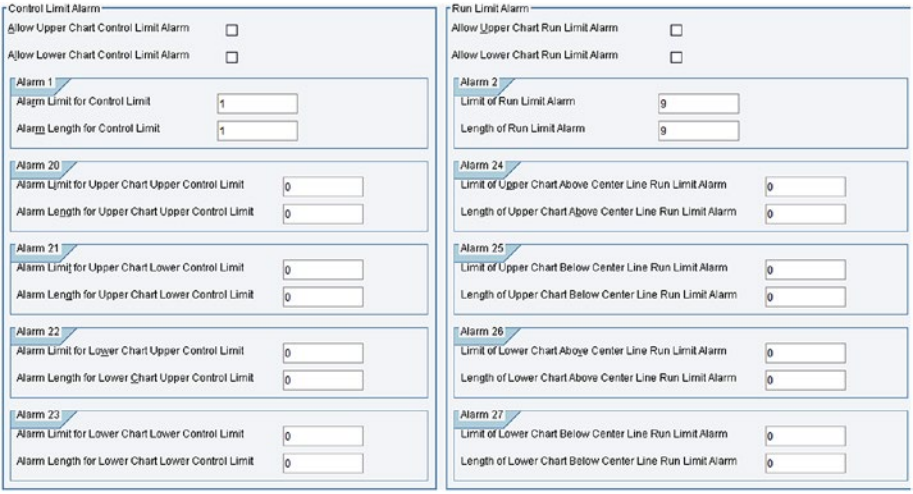


Figure 7-60. *i5SPC chart alarm configuration*

I5SPC Chart Output

Figure 7-61 shows the output for the i5SPC chart display.

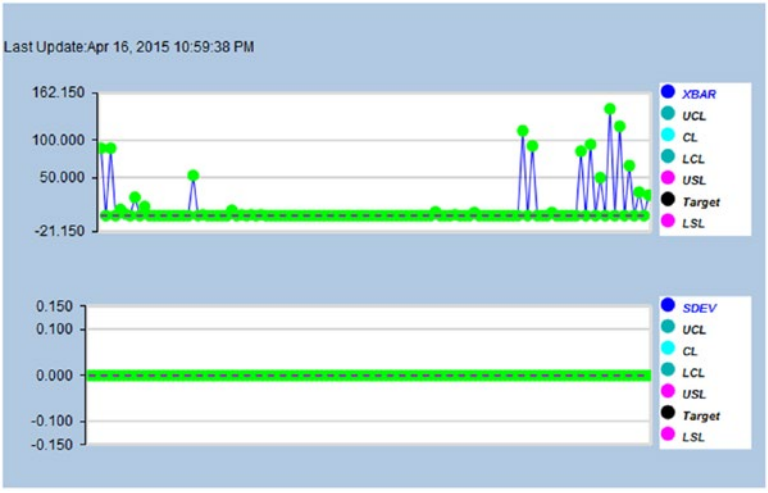


Figure 7-61. *i5SPC chart sample output*

It is possible to create different kinds of i5SPC charts like XBAR, Histogram, etc., as shown in Figure 7-62.

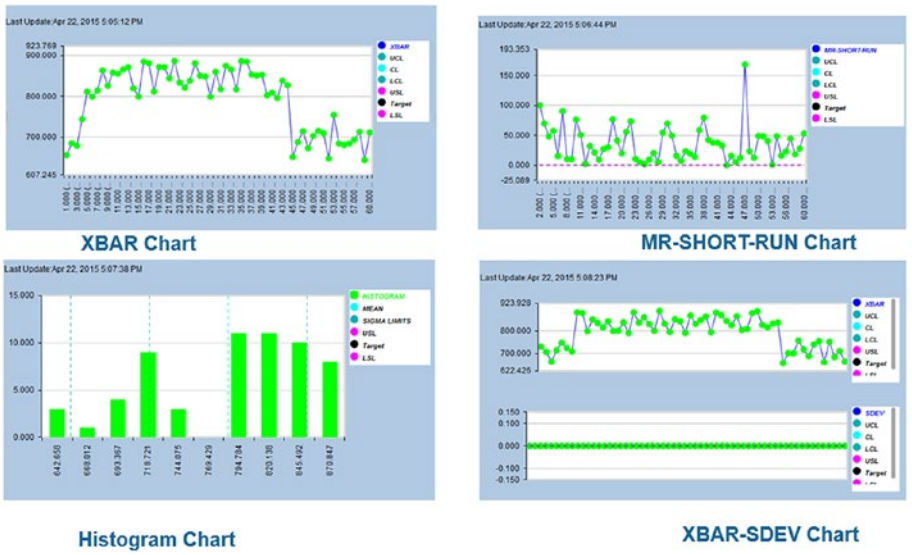


Figure 7-62. *i5SPC chart samples with various chart types*

i5SPC Chart: Rendering Options of i5SPC Charts in UI

The code to include the i5SPC chart in an HTML page is as follows:

```
var i5SPCChart = new com.sap.xmii.chart.hchart.i5Chart("Default/Saptaparna/
i5SPCChart", "Default/Saptaparna/TagQuery);
i5SPCChart.setChartWidth("400px");
i5SPCChart.setChartHeight("400px");
i5SPCChart.draw(“SPCchartDiv");
```

i5SPC Chart Features

The following are some features of the i5SPC chart:

- *Control limits and specifications limits*: Specification limits are external tolerance values set by the user/customer and apply to individual units. Control limits are horizontal lines on a control chart that represent borderlines for a process.

- *Control limits*: Control limits indicate the behavior of the process and whether the process is in control. The upper control limit (UCL) and lower control limit (LCL) are calculated statistically from the data available in the queries. UCL and LCL are set by calculating the mean, standard deviation, and range of process data collected for a stable process. See Figure 7-63.

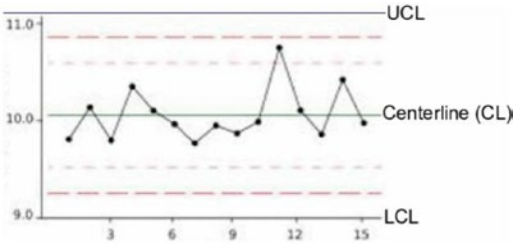


Figure 7-63. *Control limit and specification limit of i5SPC chart*

- *Specification Limits*: Specification limits determine process capability and sigma values. Upper specification limits (USL) and lower specification limits (LSL) for each chart can be defined. Control limits are limits or averages that are calculated based on previous history.

  a) *Subgroup*: An important aspect of SPC chart is organizing the data collection into subgroups. Each subgroup must be in control. Data points are either averages of subgroup measurements or individual measurements plotted on the Label and Value columns.

  b) *Warning Limits*: Warning limits on control charts are limits that are inside the control limits. Upper warning limits and lower warning limits can be defined. When warning limits are used, control limits are referred to as action limits. By default, warning limits are set at 2-sigma. Advantages of using warning limits are that they signal process changes more quickly than the 3-sigma action limits. WECO rule is considered for a process when two of the three data points are between 2-sigma and 3-sigma of the chart centerline.

In control charts:

- If the mean lies within warning limits, no action is taken.
- If the mean lies between warning and action limits, take another sample.
- If the mean lies outside action limits, take action.

c) *Inner Limits*: Inner limits are set at +/- 1 sigma distance from the centerline. You can set inner limits for upper and lower charts.

i5ValueHelp

i5ValueHelp provides the feature of filtering the correct input data from any table. If value help is used in any table, when the user starts typing the words or part of a word in the text field of Value help, the table is filtered instantly and returns any matches. To use Value help, a query template is mandatory.

In the i5Value help configurations, four properties can be configured:

- General
- Search Results
- Search Criteria
- Template Security

Animated Objects

Animated objects are the customized User Interface (UI) widgets for applications such as digital cockpits or dashboards, label generation, specialized UI, and report components, which you can use without applets or other controls.

These objects are created in the SAP MII Workbench and are based on Scalable Vector Graphics (SVG), an XML-based format for representing vector drawings.

SVG files and animated objects are transformed into image files using SVG Renderer and Animation Renderer actions under Business Logic Transactions in the SAP MII Workbench.

Configuring an Animated Object

Follow these steps to create an animated object:

1. An animated object can be created using the File ➤ New ➤ Animated Object menu option in the SAP MII Workbench.
2. Once an animated object has been created, the various options for configuring it become available in the bottom-left panel of the Workbench, as shown in Figure 7-64.

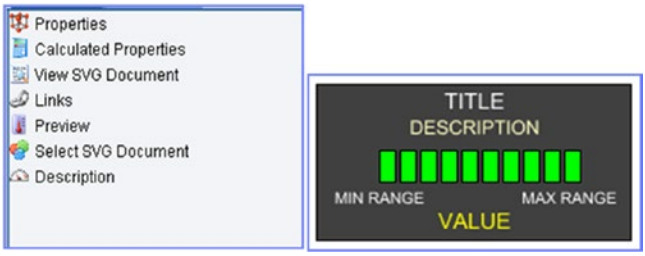


Figure 7-64. *Configuration options and an example of an animated object*

Selecting a SVG Document

You can select an SVG document, which provides the animated object definition to render the image. See Figure 7-65.

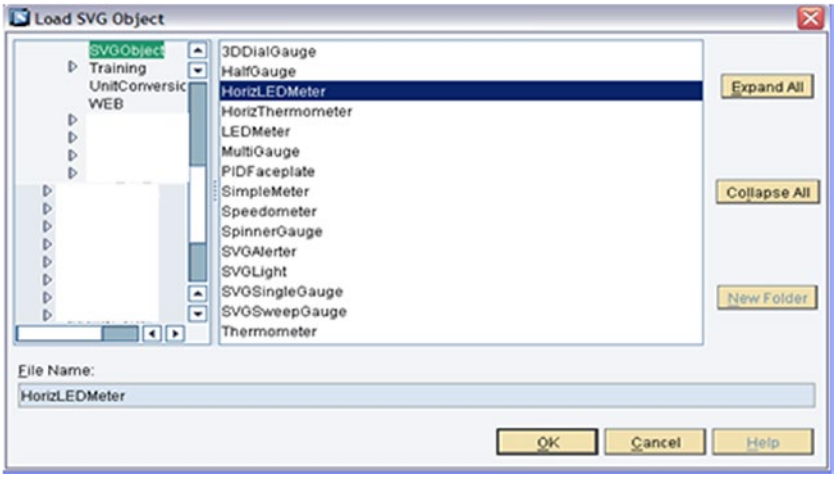


Figure 7-65. *Various types of animated objects that support SAP MII*

Properties of Animated Objects

You can view a list of all the public properties of the animated object. Public properties are object properties of the animated object that can be accessed and manipulated from a BLS transaction, as shown in Figure 7-66.

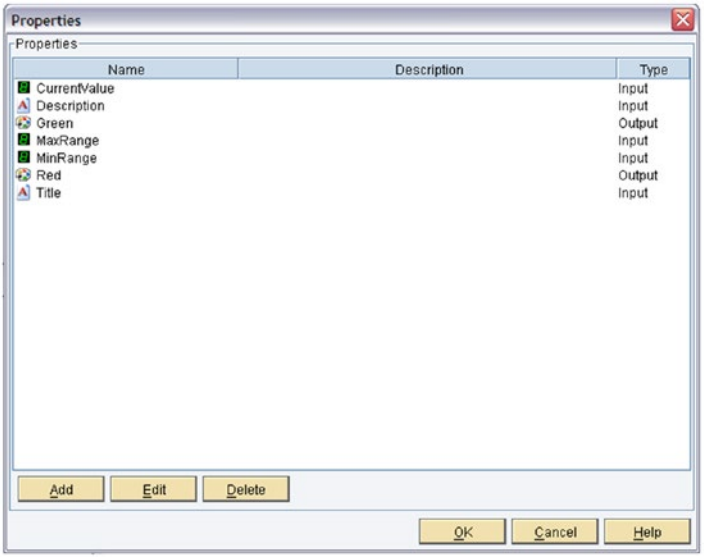


Figure 7-66. *Public Properties of animated objects*

From the public property configurations, a user can create or update any public property. Other than that, User also can define some condition or calculations under calculated properties for that same animated object. If user wants, they can check the XML beneath the object. User also can manipulate Internal and external properties of Object to manipulate the animation and once done user can preview the final animated object.

Action Blocks for Animated Objects

In MII Workbench, MII provides this set of action blocks used to render an animated object as an image in a web page:

- *SVG Renderer*: Used to take a SVG document and generate an image that represents the rendered drawing in a supported image format.
- *Animation Renderer*: Used to generate a SVG document with animation applied to the image.
- *Image Combiner*: Used to combine two generated images.
- *Image Creator*: Used to create an image with specified properties.
- *Image Scaler*: Used to scale an image to a certain height and width.

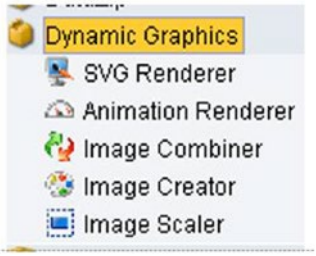


Figure 7-67. *Action block to handle animated objects in SAP MII*

Figure 7-68 shows the code sample needed to view animated objects on a web page.

```
<html>
<head>
<title>Dynamic Graphics</title>
<script language="javascript">
var Time;
function doit(){
imagename = "/XMII/Runner?Transaction=                    Dynamic
Graphics/SpeedoMeterTransaction&OutputParameter=ImageOut&content-type=image/png";
tmp = new Date();
tmp = "?"+tmp.getTime()
document.images["refresh"].src = imagename+tmp;
Time = setTimeout("doit()", 1000);
}
</script>
</head>
<body onload="doit()">

<img src="/XMII/Runner?Transaction=                    'Dynamic
Graphics/SpeedoMeterTransaction&OutputParameter=ImageOut&content-type=image/png"
name="refresh">
</body>
</html>
```

Figure 7-68. *Code sample to view animated objects in a web page*

SAP UI5

In SAP MII (14.0 and above), it's best to use the SAP UI5 frontend technology. It can be used in earlier versions as well, but the complete UI5 library must be loaded separately in the Workbench. SAP UI5 (SAP user interface library needs to be or HTML 5) is a collection of libraries that developers can use to build desktop and mobile applications that run in a browser.

SAP UI5 provides rich Internet applications, as it based on a cross-browser JavaScript library. SAP UI5 is a lightweight programming model consisting of standard and extension controls. Additionally, you can create new custom controls or extend the existing ones.

i5Charts, i5Grids, and i5SPC charts use SAP UI5's library to represent data that is obtained from the data sources graphically. The SAP UI5 application for the SAP MII environment can be developed following the MVC (Model View Controller) pattern. See Figure 7-69.

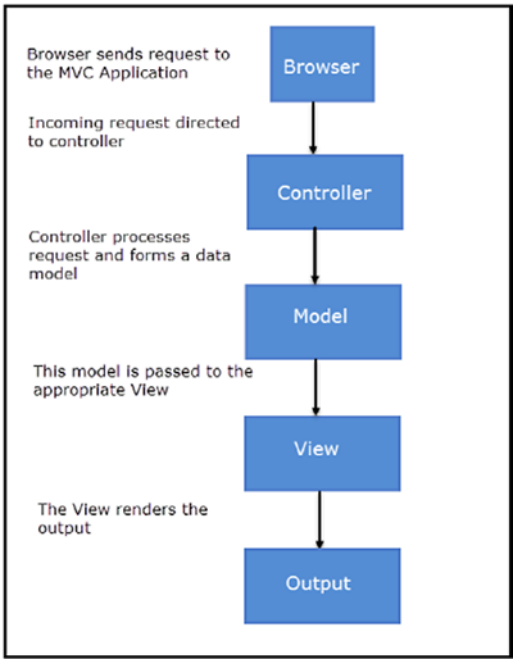


Figure 7-69. *The SAP UI5 execution model*

■ **Note** Model View Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components is built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects.

MVC consists of three concepts. Views can be defined using different languages like JavaScript or HTML. Controller binds the views and models used with views.

Views can be defined using XML with HTML, mixed, or standalone:

- *XML*: `sap.ui.core.mvc.xmlview`
- *JavaScript*: `sap.ui.core.mvc.JSView`
- *JSON*: `sap.ui.core.mvc.JSONView`
- *HTML*: `sap.ui.core.mvc.HTMLView`

Controllers are bound to a view. They can also be used with multiple views. In the Model part, data binding can be used on the views. See Figure 7-70.

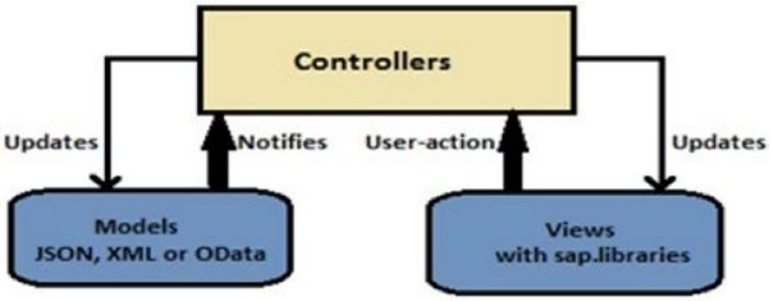


Figure 7-70. *SAP UI5 data binding*

The SAP UI5 application can be developed using different views:

- JS View
- XML View
- JSON View
- HTML View

XML view is newly introduced and is mainly used for Fiori like the Frontend using SAP UI5 and can be deployed in the Fiori Launchpad. SAP Fiori is a new user experience (UX) for SAP software and applications. It provides a set of applications that are used in regular business functions, like work approvals, financial apps, calculation apps, and various self-service apps. SAP Fiori enables multiple device applications that allow users to start a process on their desktop/laptops and to continue that process on a smartphone or tablet. SAP has developed Fiori apps based on the UI5 user interface.

SAP UI5 (SAP user interface for HTML 5) is a collection of libraries that developers can use to build desktop and mobile applications that run in a browser. With SAP's SAP UI5 JavaScript toolkit, developers can build SAP web applications using HTML5 web development standards.

SAP UI5 supports all the key web browsers and latest versions like IE, Mozilla Firefox, Google Chrome, and Safari.

SAP UI5 Data Binding

Data binding is used by the UI5 controls to hold the application data. It allows you to change the controls automatically whenever there is a change to the application data. When two-way data binding is used, application data is updated whenever the value of a bound control changes.

Data Binding Model Types

SAP UI5 supports three types of model implementation:

- *JSON Model*: Supports data in JavaScript Object Notation format. It supports two-way data binding.
- *XML Model*: Supports XML data. It supports two-way data binding.
- *OData Model*: Creates OData requests and handles responses accordingly. It only supports OData-compliant data. It supports experimental two-way data binding.

SSCE

SAP MII has launched the Self-Service Composition Dashboard, which is integrated within the SAP MII package. Using SSCE, you can develop frontend screens easily by dragging and dropping UI components and binding corresponding data to them. Moreover, in the backend, the UI code is generated and can be modified at any place for the required output.

It creates, designs, configures, and displays complete dashboards as per the requirements. It is a role based application. To access the SSCE components, the `XMII_SSCE_ALL` role must be assigned to the user. Main SSCE features and functionalities are available::

- *Dashboard creation*: Using any SAP MII content (Query Templates, Display Templates, MDO/KPI Objects, and Resource Files), UI elements, and tags from Plant Information Catalog, dashboards can be created.
- *Configuration of selected objects*: The selected tags from Plant Information Catalog (PIC) can be positioned on images. Colors, graphics, and icons can be configured as per the required conditions for the selected tags. A tag trend (line chart) is displayed when a tag is moved to an empty cell on SSCE.
- *Flexible layouts*: This displays data from any query template and user specific UI controls using different layouts of SSCE.
- *Flexible dashboards*: This saves the created dashboard for later use and so if required it can be modified easily as needed.
- *Customization*: This customizes the created dashboard by adding the saved content to the navigation tree.
- *Preview of dashboards*: This previews the created dashboard with either static or live images.
- *Security*: This secures your dashboards by assigning roles.

The three main components are Design, Preview, and Source Code. Figure 7-71 shows the design dashboard.

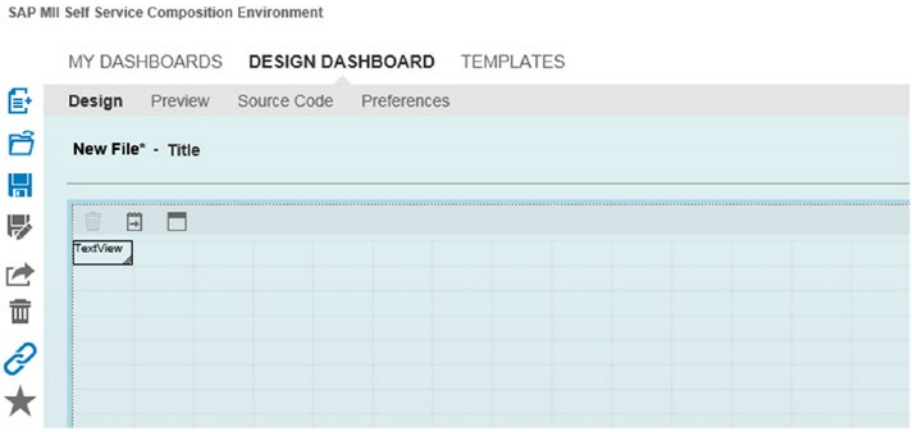


Figure 7-71. *SSCE design dashboard*

- *Design*: This is mainly used to create a dashboard using standard layouts while selecting respectable content, such as the following:
  - *MII Content*: Display Templates, Query Templates, MDO/KPI Objects, and Resource Files are the predefined objects for this section. See Figure 7-72.

Figure 7-72. *SSCE MII content menu*

- *UI Elements*: SSCE provides features to create more interactive pages by adding UI elements such as labels, text input fields, buttons, images, and drop-down boxes. These elements can be directly dropped and dragged in the Design dashboard. See Figure 7-73.

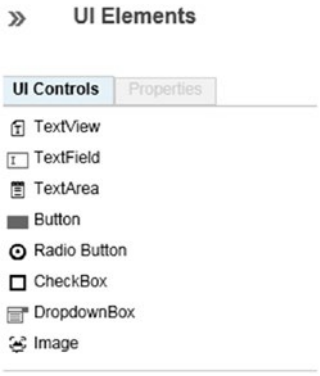


Figure 7-73. *SSCE UI elements menu*

- ✓ A reference grid is created in the background of the layout and the first UI element is always placed at the top-left cell of the grid. The consecutive elements are placed at the cursor position of the grid.
- ✓ Once the selected UI element is chosen, the Properties tab displaying all the properties associated with that UI element appears on the right pane, as shown in Figure 7-74.

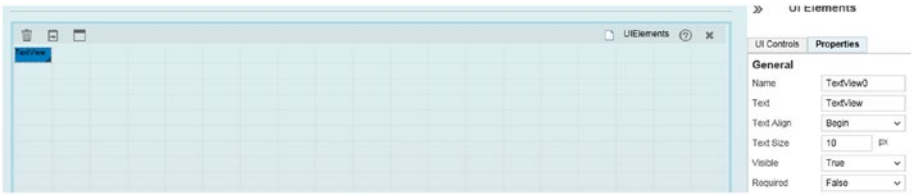


Figure 7-74. *UI element properties in SSCE*

Properties of the UI elements like ID, text value, visibility, event handlers, and so on can be configured.

- ✓ Upon adding an EventHandler, a JavaScript method is created in the custom code area in the Source Code tab. The UI elements along with the configurations in the Preview tab can be seen. See Figure 7-75.

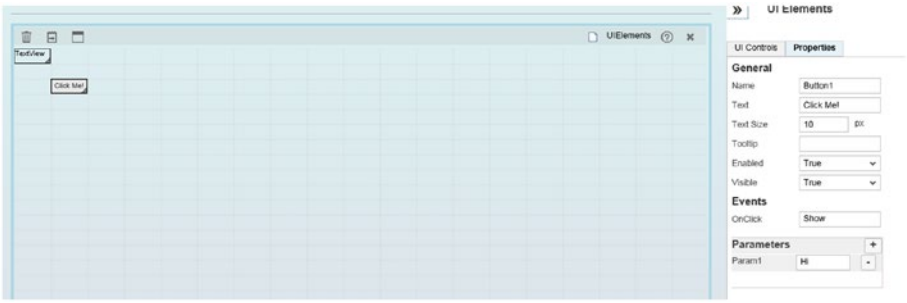


Figure 7-75. *Setting events of an UI element*

Once the Source code tab is clicked, the auto-generated code of the event handler is shown, as in Figure 7-76.

Figure 7-76. *Event handler auto-generated code in SSCE*

- *Plant Information catalog*: A tag from the Plant Information Catalog can be moved to an empty cell, on to an image, or on a tag trend. It can be repositioned to the desired location. The selected tag can be viewed in trends and current views in live and design modes. The changes/configurations to the tag are automatically captured in the Preview mode.

Customized dashboard layouts can be searched and opened from the Open Dashboard command. Moreover, content that you create can be configured and previewed in the Preview tab.

Merging, Unmerging, and deleting the dashboard are also possible. A live mode of dashboard can also be viewed as per requirement. Navigation links to the dashboards can also be created. Security is an added advantage.

The UI Dashboard can be seen in the Preview tab, as shown in Figure 7-77.

Figure 7-77. *Preview window in SSCE*

The Source Code tab can be used to view the generated HTML source code for the created dashboard. External code can be appended to the existing source code with additional CSS or JavaScript from the SAP MII Workbench. JavaScript is added to the generated source code in the section that is marked for a user-entered script.

When you choose Edit, the entire source code area is split into two parts, as shown in Figure 7-78.

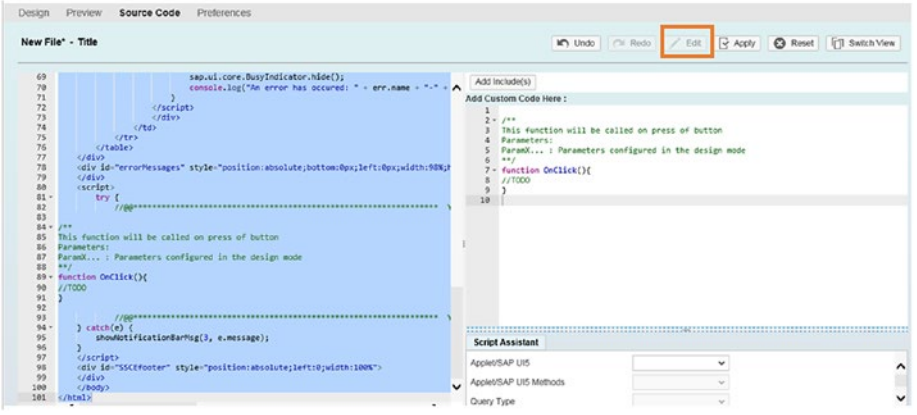


Figure 7-78. *Source Code editor in SSCE*

- The left part represents the generated HTML source code, which is not editable.
- The right part represents the hook points that can be edited. This section is further split into two areas: Custom Code and Script Assistant.

In the Custom Code area, you can add code that must be available at runtime. You can use any objects that are created in the generated code area or the available APIs to modify the behavior of the objects at runtime using Add Include(s).

In Script Assistant, you can further improve the scripting syntax by providing available options with the applet, its methods, Query Type, and Query methods, to bind with the applet. The following configurations are available (see Figure 7-79):

- *Applet/SAP UI5*: Lists the applet names along with the SAP UI5 objects.

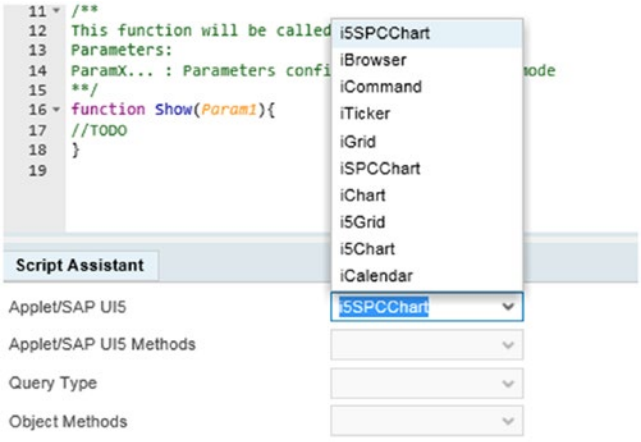


Figure 7-79. *Available UI5 applets in SSCE*

- *Applet/SAP UI5 Methods*: Lists the base applet names along with SAP UI5 object methods. See Figure 7-80.

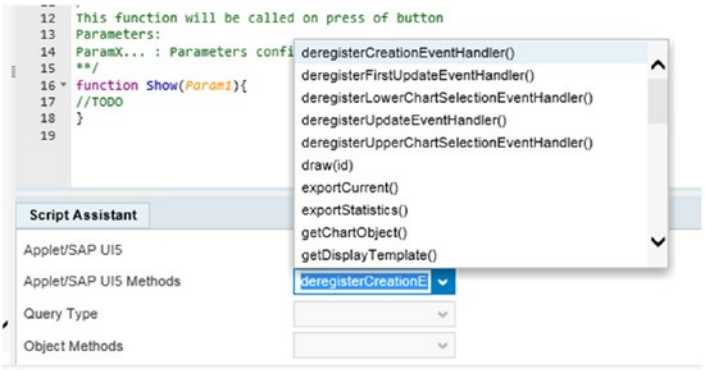


Figure 7-80. *Available UI5 methods in SSCE*

- *Query Type*: Lists the query type if any associated with the Applet method being selected. See Figure 7-81.

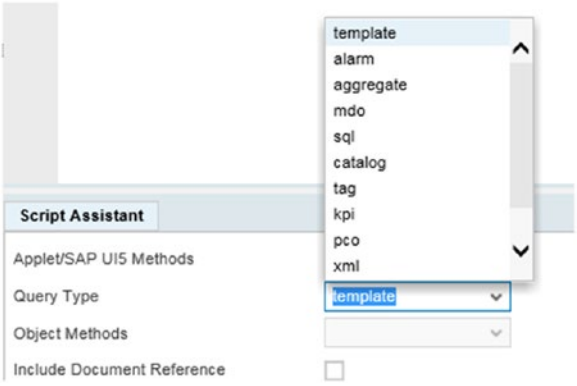


Figure 7-81. *Available query types in SSCE*

- *Object Methods*: Lists the object method names that apply to the selected applet/SAP UI5 object. See Figure 7-82.

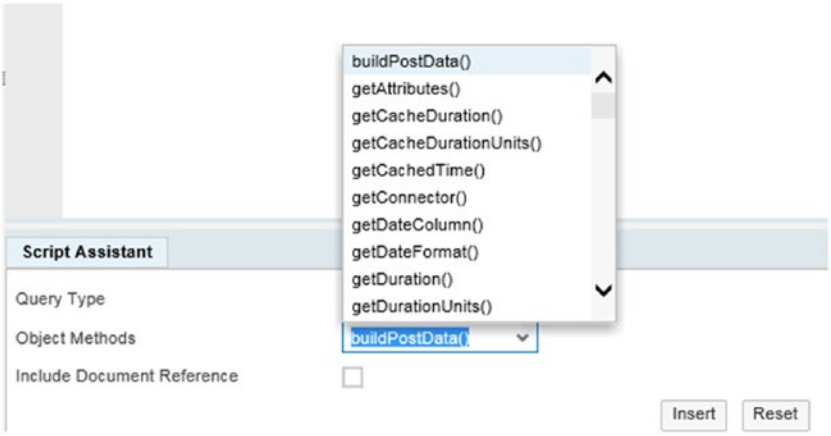


Figure 7-82. *Available object methods in SSCE*

- *Include Document Object Reference*: If the checkbox is selected, the system adds the document object as a prefix to the applet name to reference an applet or a control on the web page. It is recommended to use a prefix. For example, when the checkbox is checked, a document prefix reference is added (see Figure 7-83).

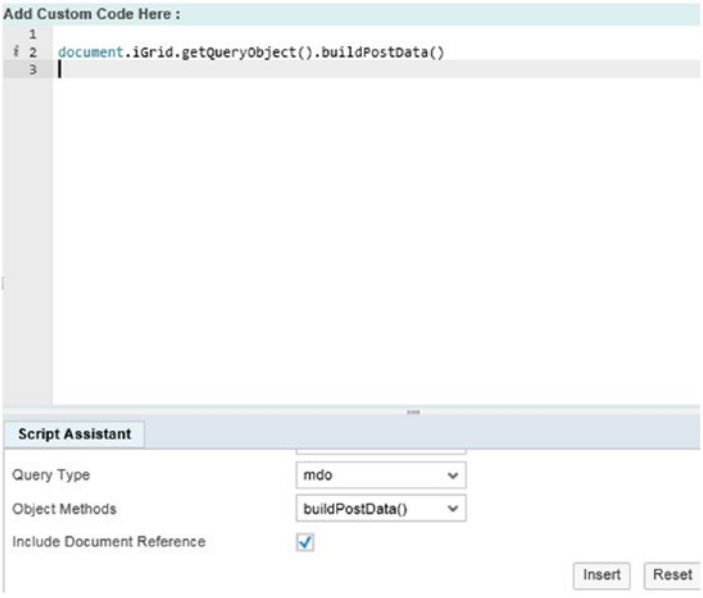


Figure 7-83. *Including a document reference in SSCE*

If the checkbox is not selected, no reference to the document prefix is created, as shown in Figure 7-84.

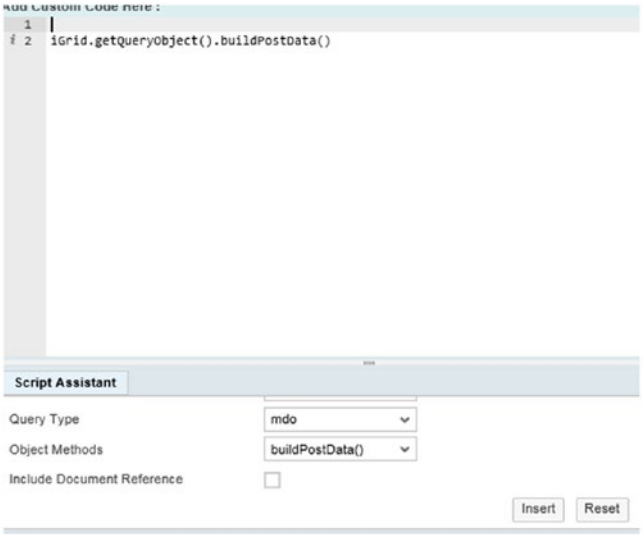


Figure 7-84. *Excluding a document reference in SSCE*

Key Performance Indicators (KPIs)

Key Performance Indicators (KPI) are numerical index measures that monitor critical operational areas and let the business know how well the area or the organization is performing. KPIs are critical for performance evaluation. KPIs are generally configured for specific business scenarios and to let the associated management know whether areas are performing well, need monitoring, or need immediate attention and improvement.

KPIs can be created and configured in SAP MII and can also be monitored using the built-in KPI monitoring screen.

The KPI Framework

The KPI framework is a built-in feature of SAP MII that allows you to define KPIs and create logic to calculate and monitor. This framework also allows you to create custom UIs and consume the KPI data from SAP MII. It can fetch required data for KPI calculation from any shopfloor system or historians using data connectors. Custom logic creation is also possible for KPI calculation.

Here are the major steps of the KPI framework:

- KPI definition
- Data acquisition
- Data consumption

KPI Definition

KPI definition is where KPIs are created and configured. It needs the `XMII_DEVELOPER` role assigned to create KPI.

Type of KPIs

SAP MII has two types of KPIs:

- *Base KPIs*: Not dependent on any other KPI
- *Composite KPIs*: Dependent on different other KPIs to get data

Base KPIs

Base KPIs are independent, self-sufficient performance indicators that can collect and calculate values by themselves without the help of an external system. These base KPIs can be part of other KPIs to gather required data.

To configure base KPIs, follow these steps:

1. Open the SAP MII menu and choose Workbench ➤ Launch Workbench.
2. Open the Object tab.
3. Right-click on the Project folder.
4. Select New File ➤ Base KPI, as shown in Figure 7-85.

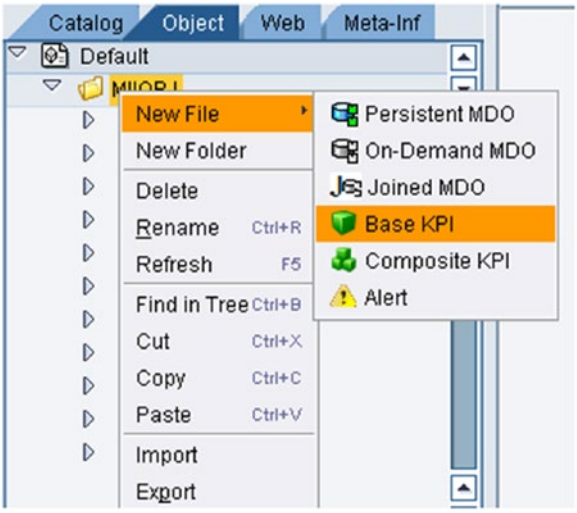


Figure 7-85. *Selecting Base KPI*

5. There are links to different configuration pages found in the bottom-left corner of the page. See Figure 7-86.

Figure 7-86. *Template categories for Base KPIs*

The General page contains the header information of the KPI. Available fields in this page are shown in Figure 7-87.

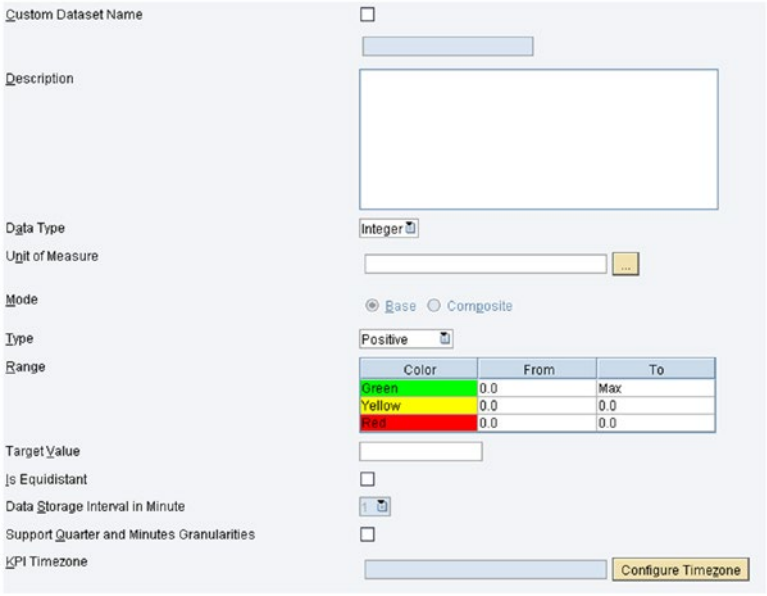


Figure 7-87. *General screen in Base KPI*

- *Description*: This specifies the description of the KPI.
- *Data Type*: This field specifies the data type of the generated KPI values. Available types are integer, double, float, and long.
- *Unit of Measure*: This specifies the measuring unit of the KPI.
- *Mode*: This is a read-only field that specifies the type of the KPI (base/composite).
- *Type*: Three available types are:
  - *Positive*: The higher value KPI indicates better performance. As shown in Figure 7-88, when the type is selected as positive, green is defined for +10 to Max KPI value, yellow is defined for +5 to +10 KPI values, and red is defined for 0 to +5 KPI values.

Figure 7-88. *Sample for positive type Base KPI*

  - *Negative*: The opposite of the positive indicator. Red falls within the range of -10 to max negative KPI value, yellow comes within -5 to -10 KPI values, and green comes for 0 to -5 KPI values. See Figure 7-89.

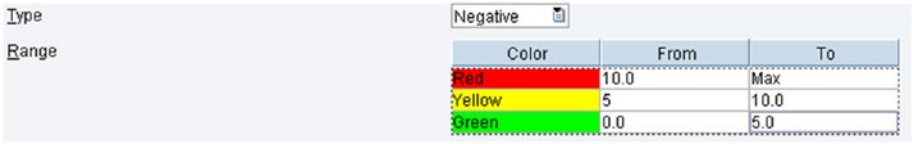


Figure 7-89. *Sample for negative type Base KPI*

  - *Bidirectional*: This means it should be in a specific range with both higher and lower limits. For this type, in Figure 7-90, red falls within the range of +20 to max and 0 to +5 KPI values, yellow falls within the range of _+15 to +20 and +5 to +10 KPI values, and green comes under +10 to +15 KPI value range.

Figure 7-90. *Sample for bidirectional type Base KPI*

Range: Depends on the type. For a positive KPI, green is at the top followed by yellow and red. It's the opposite for a negative KPI.

Aggregation Logic: This field specifies the logic to be used for aggregation of values while querying the KPIs. Available options are:

- *Min*: Returns the minimum value of the retrieved result.
- *Max*: Returns the maximum value of the retrieved result.
- *Avg*: Returns the average value of the retrieved result.
- *Sum*: Returns the Summation of all values of the retrieved result.
- *Range*: Returns the difference between maximum and minimum values.

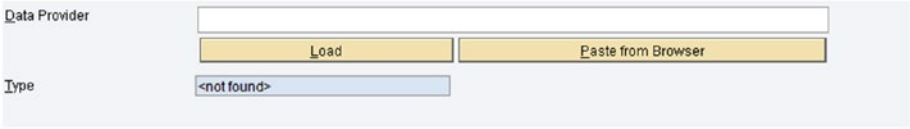


Figure 7-91. *Data provider in Base KPI*

The Data Provider page configures the data source of the KPI (see Figure 7-91). It can be a query template, a transaction, or another KPI. Click on the Load button and select the required object. Then click the OK button. Figure 7-92 shows the loading process.

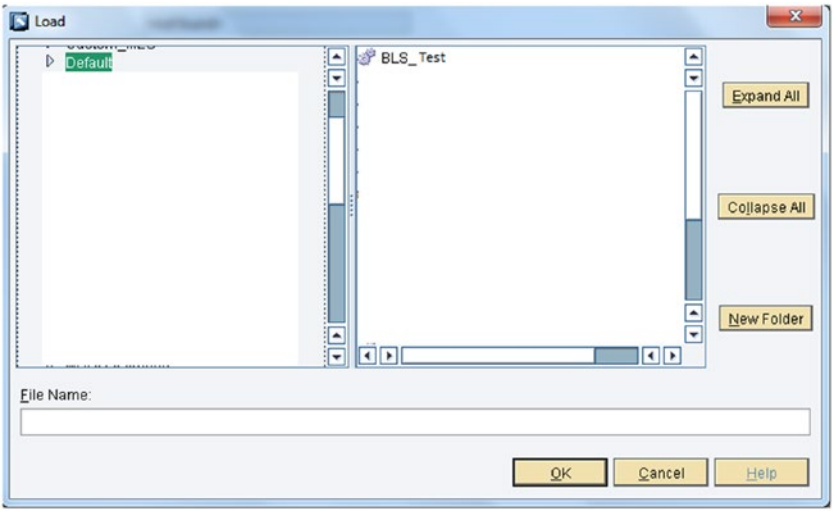


Figure 7-92. *Loading an object in the Data Provider*

Dimensions are the KPI attributes to which data coming from the Data Provider as columns can be mapped. Dimension and timestamp are two standard available attributes. Additional attributes may also be available through the Data Provider. Configure those attributes on the Dimensions page, as shown in Figure 7-93.

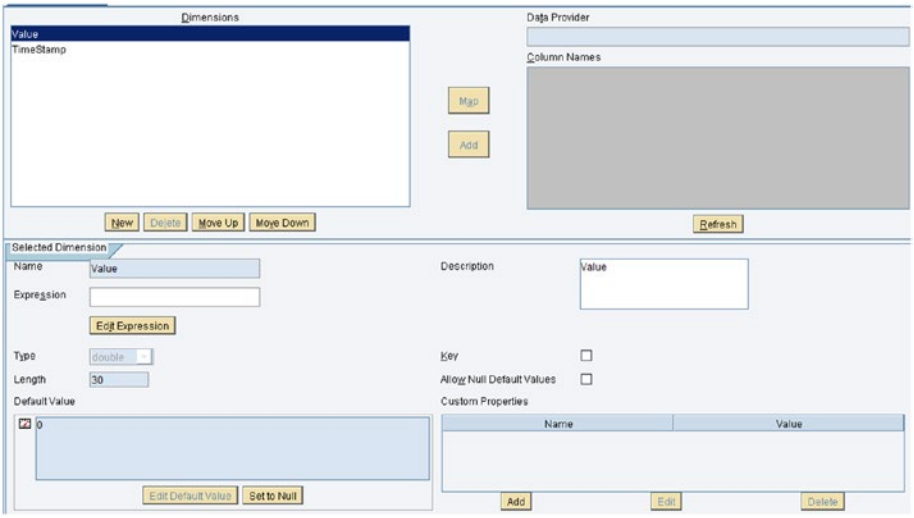


Figure 7-93. *Dimension screen in a base KPI*

Lifecycle configuration schedules the KPI to gather data at a given time. For that, a scheduler needs to be configured. Without a scheduler, KPI cannot store data. Click the Create New button and provide the mandatory information. Select the Enabled checkbox. Select a time pattern for the scheduler to run. Select the mode for the scheduler—Append, Insert, Update, or Delete. Click the Deploy All button to deploy the scheduler. It will create the scheduler itself and it will start running.

Assign the roles to the KPI to ensure that only the authorized users will be able to consume the KPI. Then save and deploy the KPI.

Composite KPIs

A composite KPI is calculated depending on the values of each of its dependent KPIs or systems. Composite KPIs may be dependent on base KPIs, composite KPIs, or both. They can be configured to collect data from multiple data sources, such as transactions, query templates, or other KPIs.

Configuring a Composite KPI

Follow these steps to configure a composite KPI:

1. Open the SAP MII menu and choose Workbench ➤ Launch Workbench.

2. Open the Object tab.
3. Right-click on the Project folder.
4. Select New File ➤ Composite KPI, as shown in Figure 7-94.

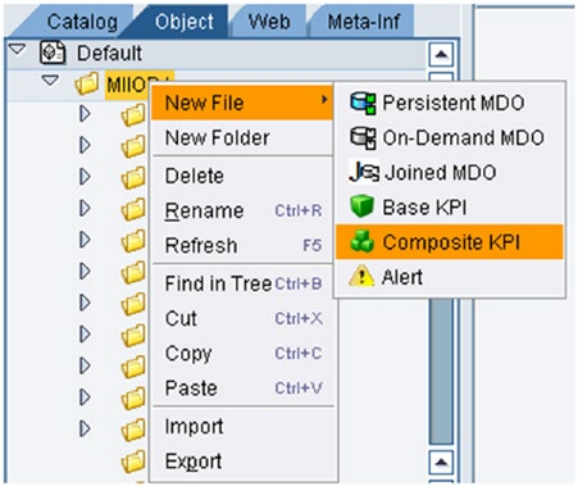


Figure 7-94. *Selecting a composite KPI*

5. There are links to different configuration pages found in the bottom-left corner of the page.
6. The General page contains the header information of the KPI. Available fields on this page are:
   - *Description*: This specifies the description of the KPI.
   - *Data Type*: This field specifies the data type of the generated KPI values. Available types are integer, double, float. and long.
   - *Unit of Measure*: This specifies the measuring unit of the KPI.
   - *Mode*: This read-only field specifies the type of the KPI (base/composite).
   - *Type*: The three available types are:
     - Positive
     - Negative
     - Bidirectional

A KPI type that's positive means a higher value KPI indicates better performance. A negative type is the opposite. A bidirectional type means it should be in a specific range with both higher and lower limits.

- *Range*: Depends on the type. For a positive KPI, Green is at the top followed by Yellow and Red. It's the opposite in the case of a negative KPI.
- *Aggregation Logic*: This field specifies the logic to be used for aggregating values while querying the KPIs. Available options are:
  - *Min*: Returns the minimum value of the retrieved result.
  - *Max*: Returns the maximum value of the retrieved result.
  - *Avg*: Returns the average value of the retrieved result.
  - *Sum*: Returns the summation of all values of the retrieved result.
  - *Range*: Returns the difference between the maximum and minimum values.

7. On the Data Provider page, there is an additional area called KPI Data Provider (see Figure 7-95). This field allows you to add the KPIs on which composite KPI is dependent. To add dependent KPIs, click the Load button and then select the KPIs from the popup box. Click OK.
8. If the values for the composite KPI are coming from an external system, the Data Provider needs to be selected at that time.

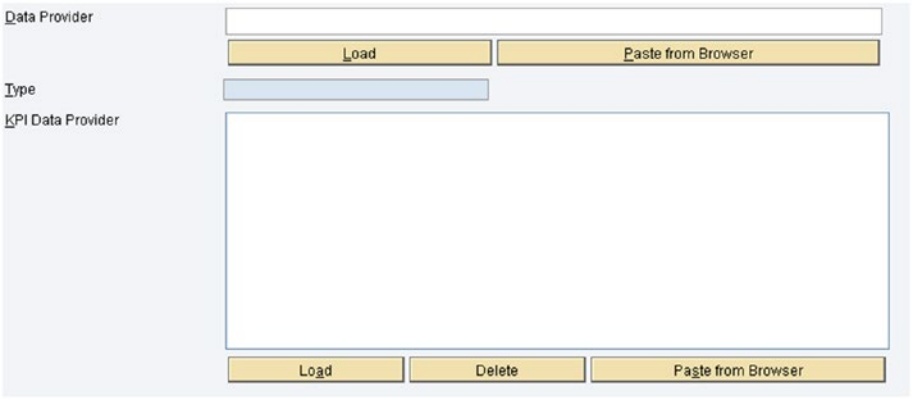


Figure 7-95. *Data Provider screen in a composite KPI*

9. The Dimension page of a composite KPI is quite similar to that of the base KPI until you try to map a value of multiple KPIs to the composite KPI. To set a value for a composite KPI, click on the Edit Expression button. Choose to edit the expression, then click OK.

10. The Lifecycle configuration schedules the KPI to gather data at a given time. For that, a scheduler needs to be configured. Without a scheduler, KPI cannot store data. Click the Create New button and add the mandatory information. Select the Enabled checkbox. Select a time pattern for the scheduler to run. Select mode for the scheduler—Append, Insert, Update, or Delete. Click the Deploy All button to deploy the scheduler. It will create the scheduler and will start running.

11. Assign roles to the KPI to ensure that only authorized users can consume the KPI.

12. Save and deploy the KPI.

Data Consumption and Monitoring

In the manufacturing industry, various parameters must be measured to monitor performance from multiple perspectives. These are called key performance indicators (KPIs). You must also monitor the data used for calculating the KPIs and the final KPI results on a regular basis. These are crucial industry requirements, as ISO certifications and quality clearance is dependent on the KPI results. SAP MII has built-in features to calculate KPI using standard KPI formulas and, if required, certain levels of custom KPI formulation can be created. This section describes how to configure and use KPIs and query data using KPI queries, as well as how to visualize that data.

KPI Query

A KPI query is a new type of query template that allows you to query KPI data.

Follow these steps to configure a KPI query:

1. Launch Workbench.

2. Go to the Catalog tab

3. Right-click on any project and select New ➤ KPI Query. See Figure 7-96.

4. The KPI Query screen has different configuration pages available at the bottom-left corner of the screen. The pages are:

   - Data Source
   - General

- Date Range
- Select Query
- Delete Query
- Insert Query
- Parameters
  - Transformation
  - Security

5. On the General page, you specify a Data Server and a connector. SAP MII provides a default data server named `KPIConnector`. This data server can be used to query the KPI values created on that server.

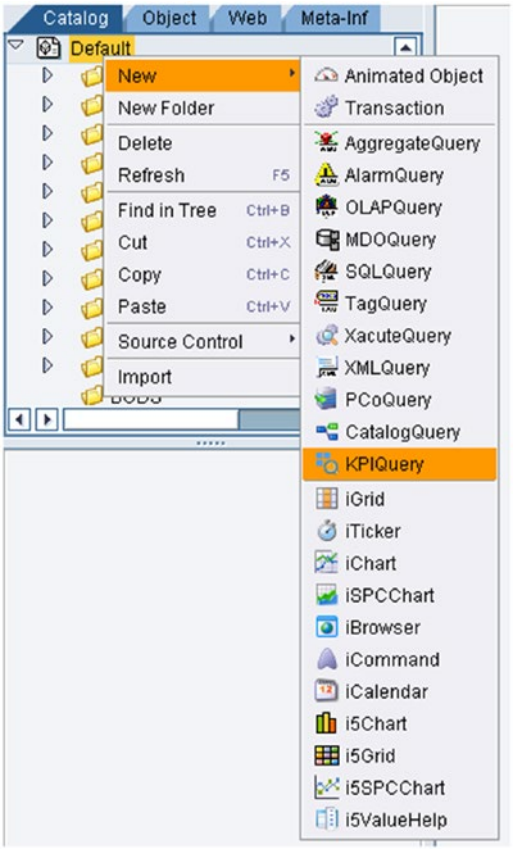


Figure 7-96. *Selecting KPI query*

6. This data server can be enabled or disabled depending on the requirements of using the server and can be maintained in the SAP MII Admin menu.
7. Click on the Load button and then select the KPI to be queried from the popup. Click OK.
8. Select the mode from the drop-down list. The available mode options are:
   - *Attribute List*: This shows the list of all KPI dimensions.
   - *Delete*: This deletes the data of the selected KPI.
   - *Info*: This shows the list of all the KPI's metadata.
   - *Insert*: This inserts data into the KPI.
   - *KPI List*: This shows the list of KPIs created and configured on the selected data server.
   - *Mode List*: This shows different modes that a KPI query can run.
   - *Select*: This retrieves data from the selected KPI.
9. Open the Dimensions page and select a dimension from the available dimensions. Choose Add. See Figure 7-97.
10. Filter expressions also can be configured from the Dimensions page.

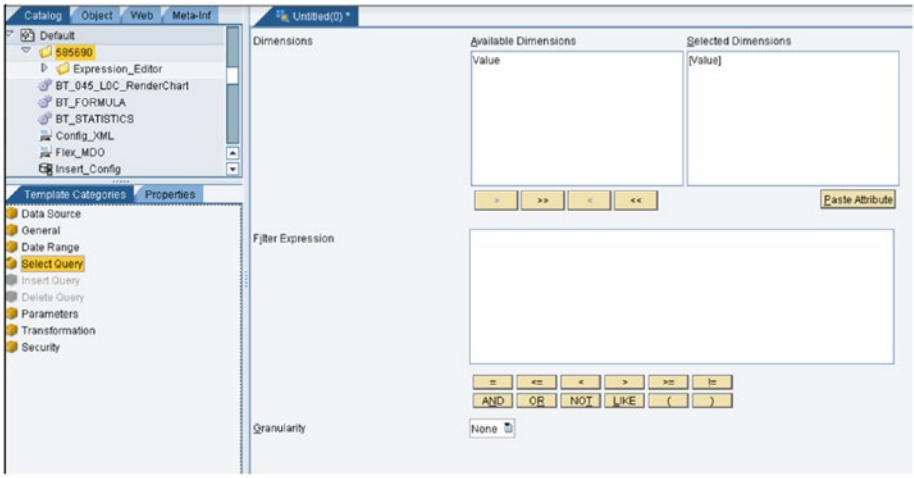


Figure 7-97. *Defining dimensions in KPI*

11. Granularity in the Dimension screen determines how the data is represented when the KPI is queried. The available granularity types are:
    - *Hour*: Data will be aggregated hourly according to the aggregation logic.
    - *Day*: Data will be aggregated daily according to the aggregation logic.
    - *Month*: Data will be aggregated monthly according to the aggregation logic.
    - *None*: Aggregation logic will be applied to the whole dataset.
    - *Raw*: It will display all the KPI data available.
    - *Week*: Data will be aggregated weekly according to the aggregation logic.
12. Save the query with a suitable name.
13. Create a display template.
14. Add the display template to the visualization page of the KPI query.
15. Execute the KPI query.

KPI Monitor Page

The KPI you just created needs to be monitored. SAP MII provides the facility to create a personalized dashboard to monitor using display templates or the KPI Monitor screen. See Figure 7-98.

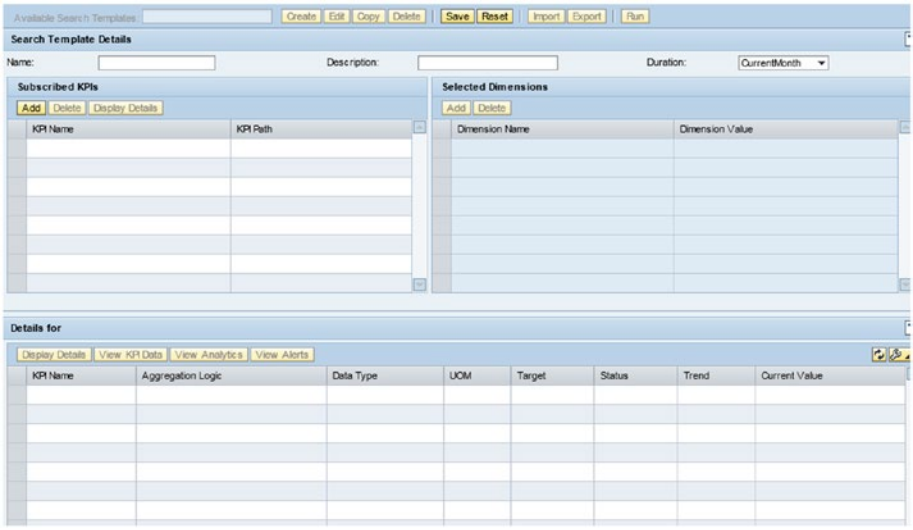


Figure 7-98. *KPI Monitor screen*

Open the KPI Monitor page from the SAP MII Admin menu by choosing the KPI & Alert category. Click on the Create button –and then choose Create Search Template with Name, Description, and Duration. Add the KPI to monitor and then click OK. Configure the dimensions of the KPI further, if needed.

Using Alerts in KPIs

KPIs can be configured to raise alerts whenever the KPI value crosses a specific limit. Multiple alerts can be configured for a single KPI.

Follow these steps to create and configure an alert:

1. Select the Object tab of the Workbench.
2. Right-click on any project folder and then choose New File ➤ Alert. See Figure 7-99.

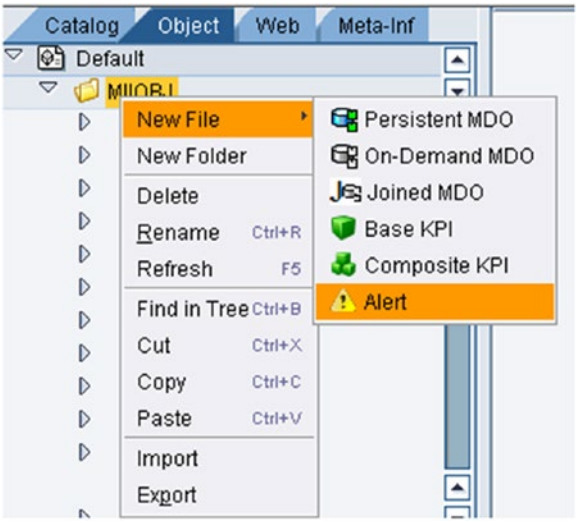


Figure 7-99. *Selecting an Alert object*

3. Add short and long text. See Figure 7-100.

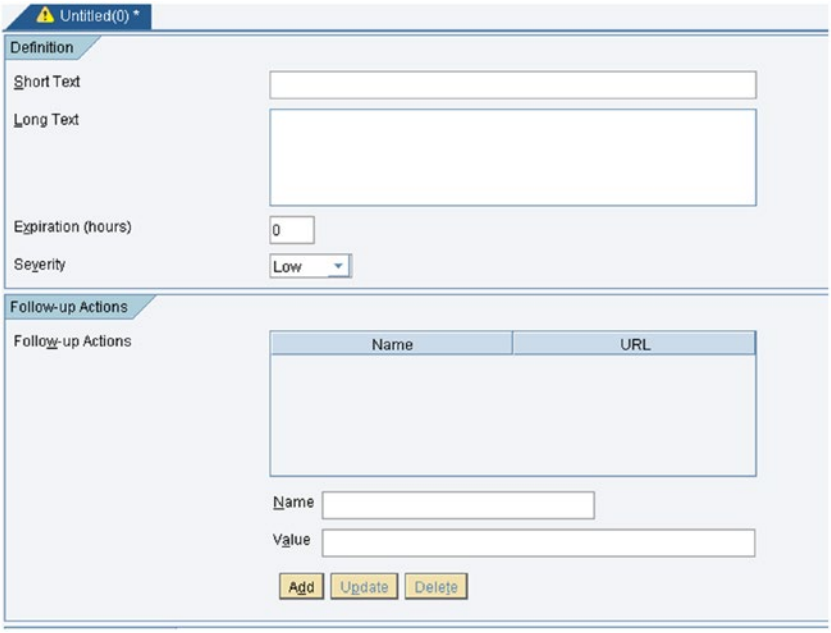


Figure 7-100. *Adding short and long text to the Alert object*

4. Add expiration hours for the alert after which the alert will automatically expire.
5. Enter an alert severity.
6. Add follow-up actions, such as the steps needed to take once the alert is raised.
7. Add any container properties that may be required to complete the alert.
8. Choose a delivery transaction that will be triggered once the alert is raised to inform the concerned users about the alert. See Figure 7-101.

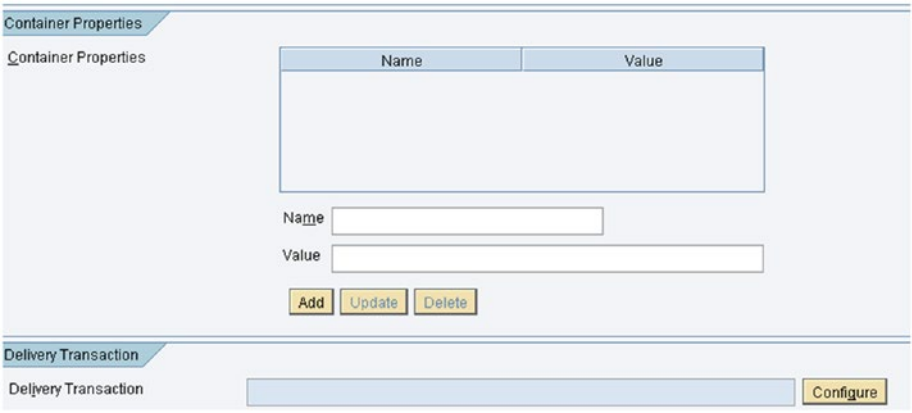


Figure 7-101. *Delivery Transaction configuration*

9. Add roles. See Figure 7-102.

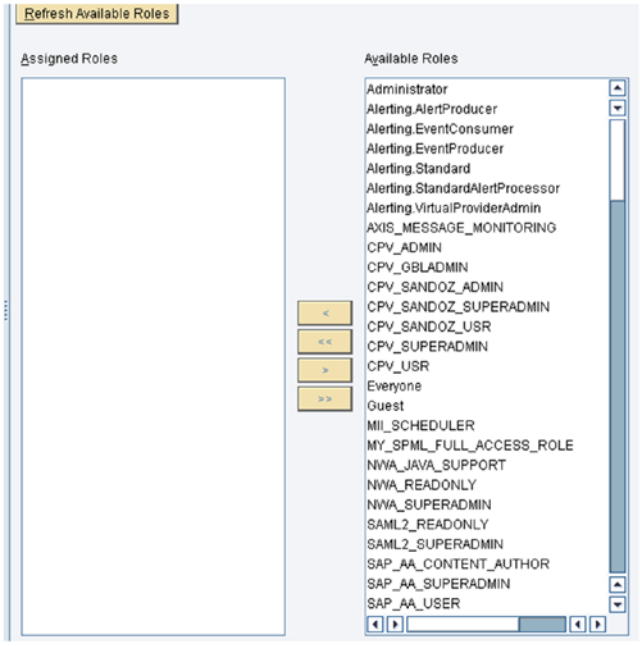


Figure 7-102. *Available security roles*

10. Save your alert with proper name.

11. Open the corresponding KPI that needs to trigger that alert and select the Enable Alerts checkbox. See Figure 7-103.

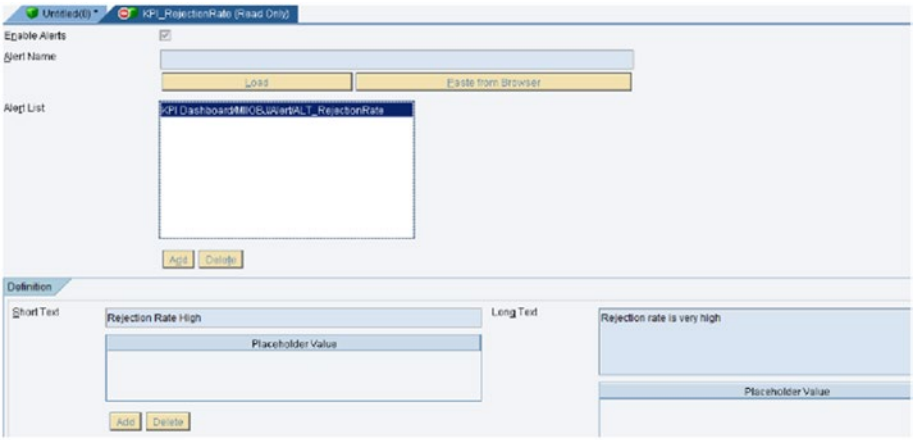


Figure 7-103. *Enabling alerts in KPI*

12. Click on the Load button in the Alert Name area and select the specific alert. Click OK.

13. Select the alert from the list and click on the Add button to add the alert to the object list.

14. Save and run.

The Plant Information Catalog (PIC)

Every manufacturing industry has data in their shopfloor systems that's stored in an unorganized manner and without any business consistency. The Plant Information Catalog (PIC) is a framework available in SAP MII for maintaining and defining the relationship of that data in an organized manner.

On the shopfloor, many systems store or capture data for different data points, such as temperature, pressure, etc., which is stored in a flat structure without any relationship to other datasets or associated business objects. The pain area for the business is that they have all the data available, but no business context associated with it. As a result, it is difficult for the business to search for specific data and analyze or use it.

Advantages of PIC

The Plant Information Catalog (PIC) has the following advantages:

- PIC provides the framework to create hierarchical, plant-specific catalogs of data points or tags from different shopfloor systems.
- PIC provides the business context to this hierarchy from ERP.
- Through PIC, shopfloor assets can be grouped logically and by business rules defined in SAP ERP systems.
- Business friendly naming conventions can be followed for the objects and data points or tags to make it easily usable to business users and developers.

PIC comes between shopfloor and enterprise systems (see Figure 7-104). It gets data from the shopfloor system and gets business context from the ERP systems and combines those into a hierarchical format to increase usability.

PIC hierarchy contains two types of objects:

- *Group*: These are logical groups containing logical tags
- *Tag*: These are data points of shopfloor systems

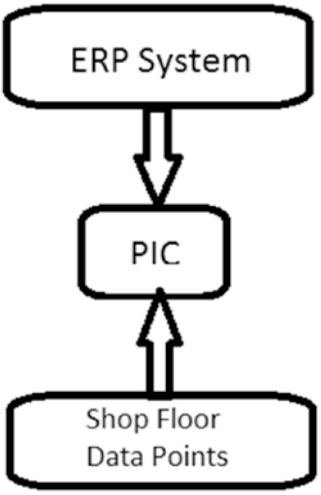


Figure 7-104. *Placement of PIC in middleware solutioning*

Creating a PIC Hierarchy in SAP MII

Creating a PIC hierarchy in SAP MII involves multiple steps, all of which are covered sequentially in the following sections.

Creating a Property Set

Property set is collection of properties that allows you to add specific values of these properties to the PIC object with which the property set is associated. Adding, updating, and deleting properties from that property set is possible from the Property Set Admin screen of SAP MII, by choosing the menu option Catalog Services ➤ Property Set. See Figure 7-105.

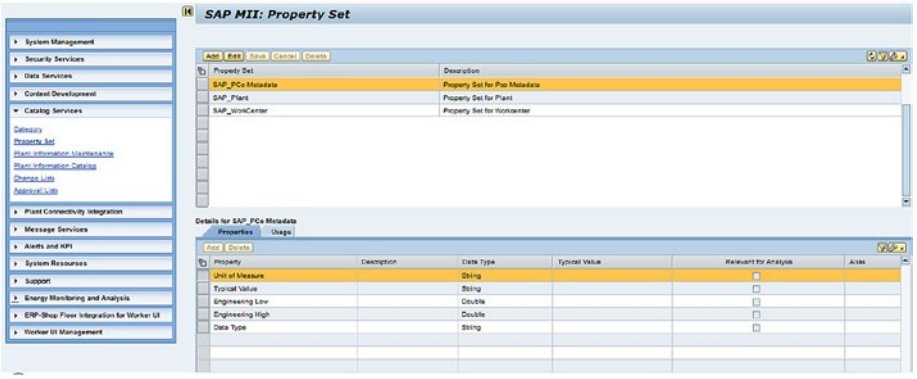


Figure 7-105. *Property Set screen in catalog services*

SAP MII includes predefined property sets that start with SAP_. These property sets contain some standard properties and cannot be updated or deleted.

The Usage tab (see Figure 7-106) gives the list of PIC objects where the selected property set is mapped. As soon as the property set is edited and saved, the system displays the list of PIC objects where the property set is mapped and asks for the choice to push the changes. If the changes are pushed, they will be reflected in the PIC hierarchy. Even if the changes are not pushed, they can be pulled from the PIC hierarchy of the Plant Information Maintenance screen.

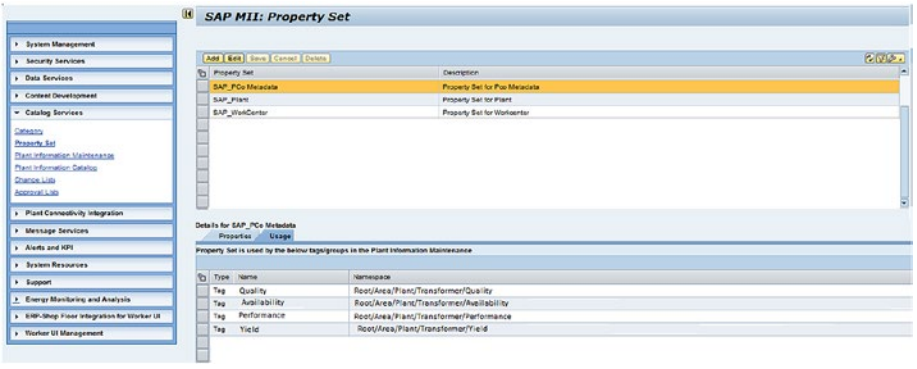


Figure 7-106. *Usage tab in Property Set*

The property set cannot be deleted if it is being used in the Plant Information Catalog.

The nodes that are not selected for updating identify themselves with an Out of Sync status and can be checked from the PIC screen for their sync status, as shown in Figure 7-107.

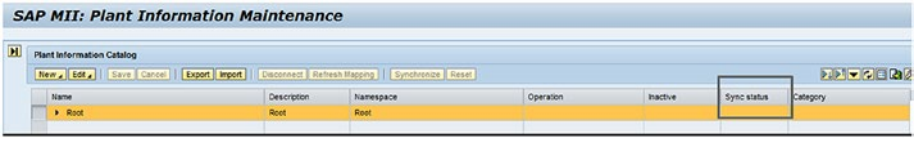


Figure 7-107. *Sync status in PIC*

Creating a Category

Categories define the types of PIC objects. As soon as a category is added to a PIC object, the property set associated with that category will also be added to that PIC object. Categories can be created, edited, or deleted from the Category Admin screen, which you can reach from the SAP MII menu path Catalog Services ➤ Category.

SAP MII includes predefined categories that start with SAP_ for property sets of functionally valid properties. These categories cannot be deleted or associated to a property set and cannot be removed. But they can be edited. Any changes to a category can be pushed to the PIC hierarchy, as per the user selection (similar to a property set). See Figure 7-108.

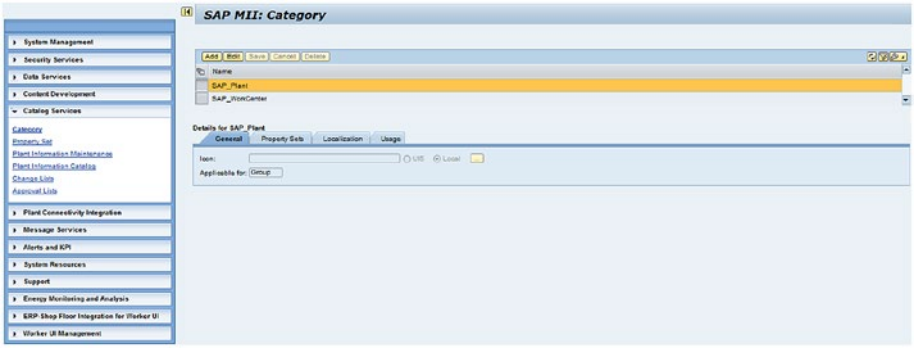


Figure 7-108. *General Category screen*

1. From the Property Sets tab, you can add a property to the Category while adding a new category or editing an existing category. See Figure 7-109.

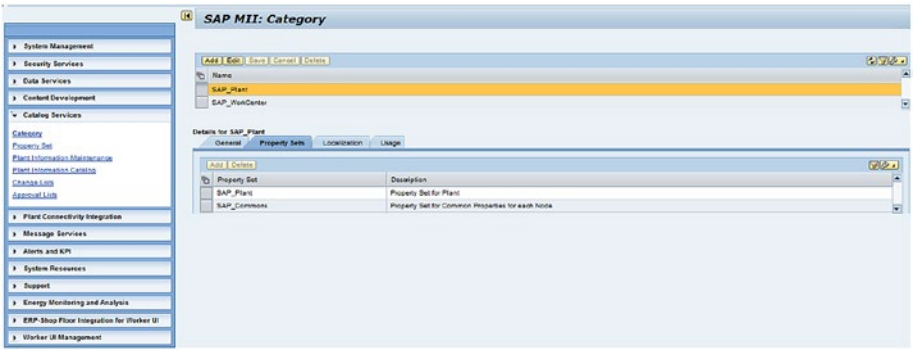


Figure 7-109. *Category screen: Property Sets*

2. In the Localization tab, different languages can be specified and, depending on the user Locale, the category description can be changed. See Figure 7-110.

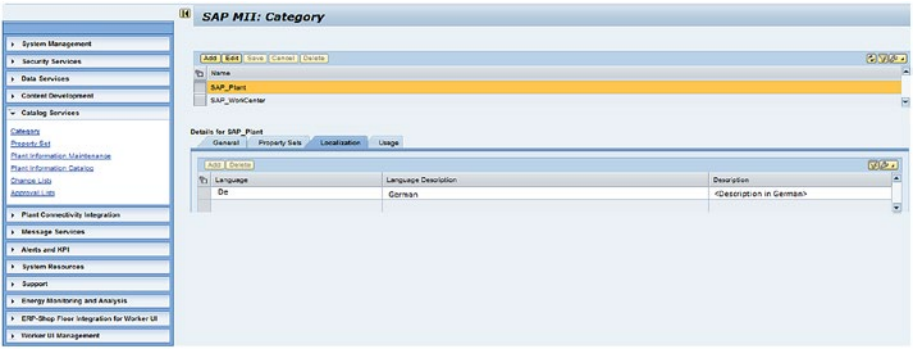


Figure 7-110. *Category Localization screen*

3. The Usage tab shows the usage of the saved categories in to the system. See Figure 7-111.

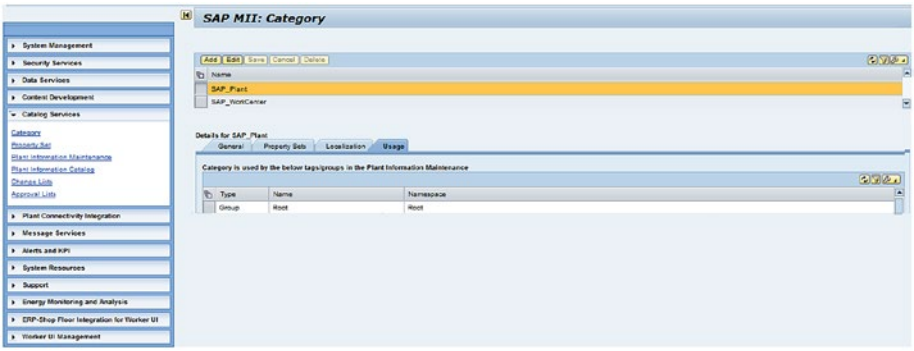


Figure 7-111. *Category Usage screen*

The category cannot be deleted if it is being used in the plant information catalog.

Creating a Change List and Marking it as the Default

Under the Catalog Services group on the SAP MII menu, there are the following two screens:

- *Plant Information Maintenance*: Plant asset hierarchy and mapping PIC objects to the actual shopfloor objects is done here. All changes are done, approved, and applied. Then the changes move to the PIC.
- *Plant Information Catalog*: This is a read-only screen for viewing the PIC hierarchy already created on the Plant Information Maintenance screen and approved.

There can be multiple change lists created by a particular user and they can remain open at the same time. Only one of them can be the current change list. Changes to the PIC hierarchy are maintained in the current change list until those changes are approved and applied to the PIC. Once all the changes to the PIC hierarchy are complete, the change list is submitted for approval. Authorized groups of users are allowed to approve or reject the change list. If it is approved, choose Approved State ➤ Apply Changes. If it is rejected, the changes are assigned back to the user for reset (the changes are reverted and the change list is closed). Figure 7-112 shows the lifecycle of a change list.

The Catalog Services ➤ Change List screen has different functionalities to work on change lists, including Create, View, Set as Current, Reset, Submit for Approval, and Delete. There is a Display Details button that gets the list of the changes made to that particular change list in a popup.

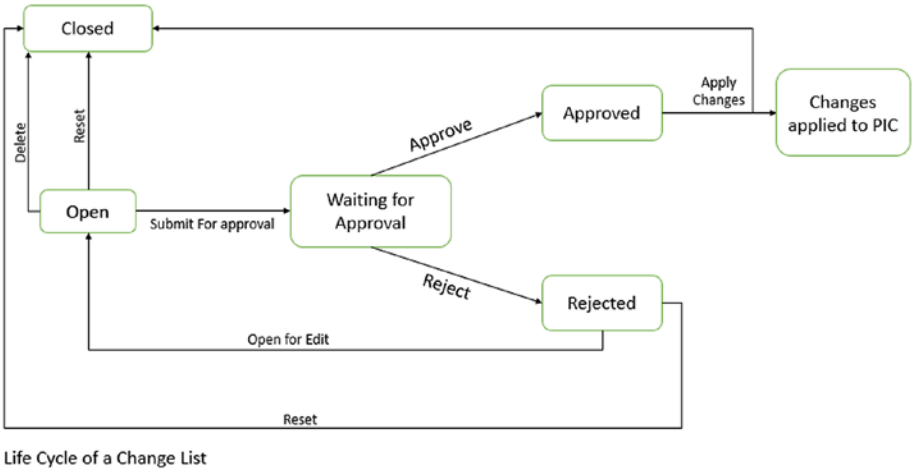


Figure 7-112. *Lifecycle of a change list*

The popup has five tabs:

- *New*: Consists of the new object changes.
- *Edit*: Consists of the edited object changes.
- *Log*: Shows the list of phases the change list has passed through.
- *Delete*: Consists of the list of deleted objects.
- *Notes to Approver*: All the extra information that an approver should know are mentioned here.

Figure 7-113 shows a created change list with a status of open.

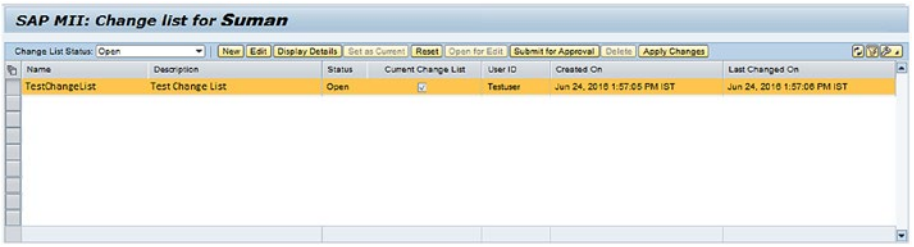


Figure 7-113. *Existing change list*

If users create a multiple change list and want to specify one of the specific change lists as the current one, they must select the change list and click on the Set as Current button to mark it. The current marked change list will appear on top of the Waiting for Approval list.

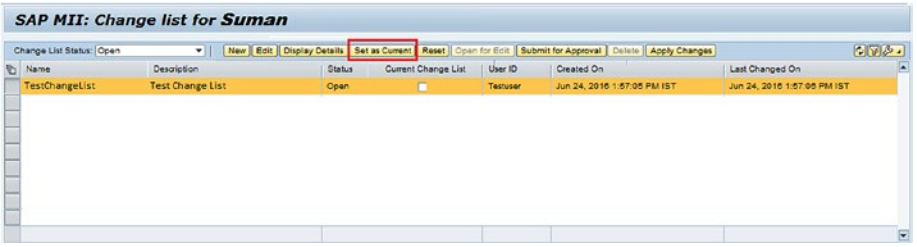


Figure 7-114. *Setting a change list as the current one*

Selecting a change list and clicking on Display Details will show the details of the particular change list. See Figure 7-115.

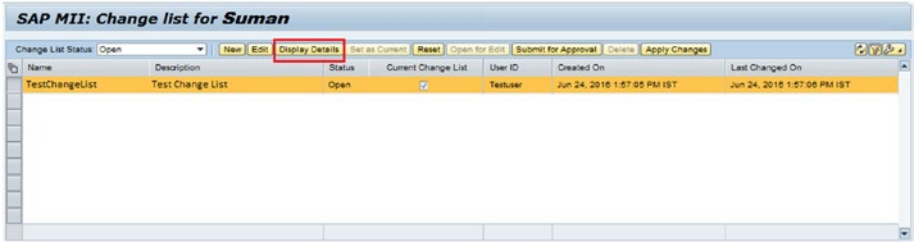


Figure 7-115. *Displaying details of a change list*

When you click Display Details, a popup will open with the new change list details. Users can either click on Submit or Reset from that screen to proceed to the next step. See Figure 7-116.

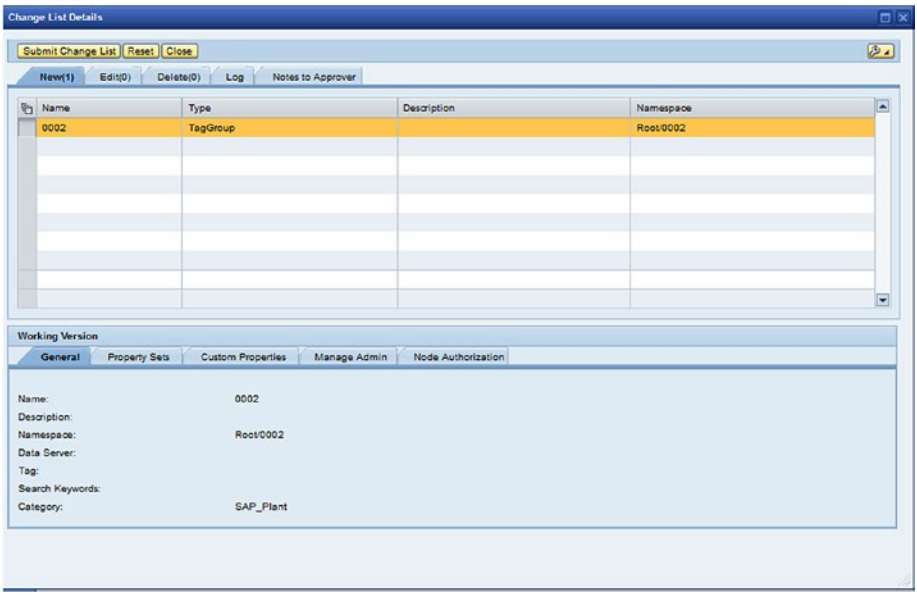


Figure 7-116. *Change List details screen*

Users can add notes to the approver, as shown in Figure 7-117.

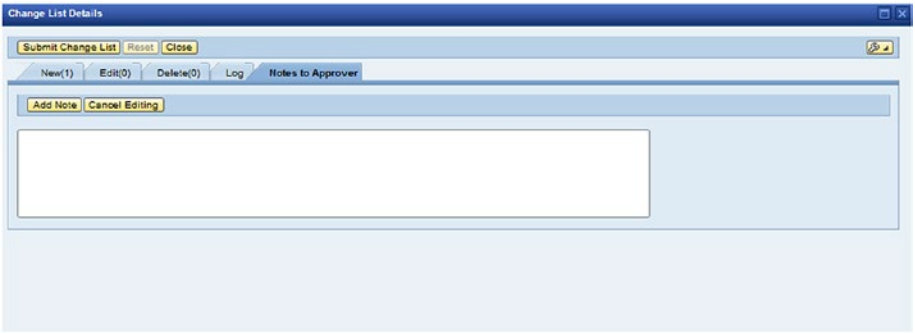


Figure 7-117. *Notes to Approver feature*

The change list is closed once it's approved and changes are applied or they are rejected and reset.

Creating a PIC Hierarchy in Plant Information Maintenance

Open Plant Information Maintenance by choosing Catalog Services ➤ Plant Information Maintenance from the SAP MII menu. The screen is divided into two parts—Plant Information Sources on the left side and Plant Information Catalog on the right side. See Figure 7-118.

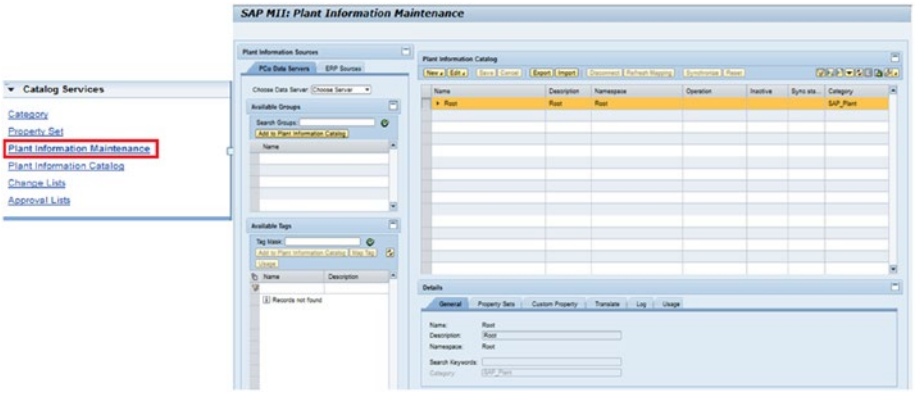


Figure 7-118. *Plant Information Maintenance screen*

At the right side of the screen, the PIC hierarchy can be created. As discussed earlier, PIC hierarchy can have two object types: Tag, which is the leaf node representing the measurement point, and Group, which are tag containers grouped logically. See Figure 7-119.

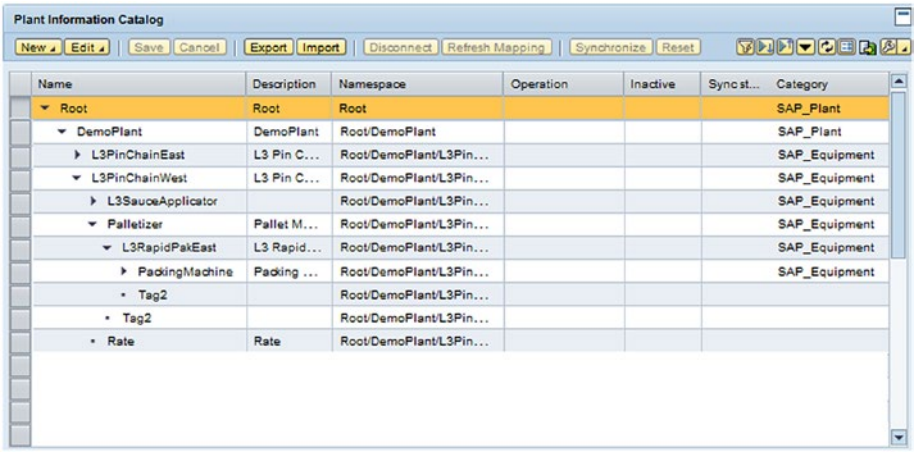


Figure 7-119. *Plant Information Catalog hierarchy*

PIC hierarchy has a default node called Root and all the created objects are located under this Root node (group). The following buttons are available at the top of the Plant Information Catalog screen:

- *New*: The options of creating a group or tag are available here (see Figure 7-120). Upon selecting the desired option, it will ask you for the Name, Description, and Category in a popup window. when you click OK, it will create the object under the selected node. *A new group or tag can be created under any group, but objects cannot be created under any tag.*

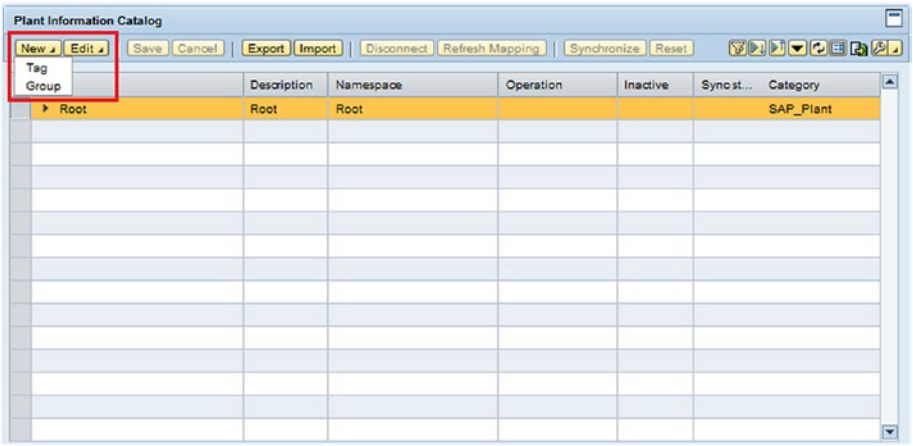


Figure 7-120. *Selecting a new PIC*

The New Group screen appears, as shown in Figure 7-121.

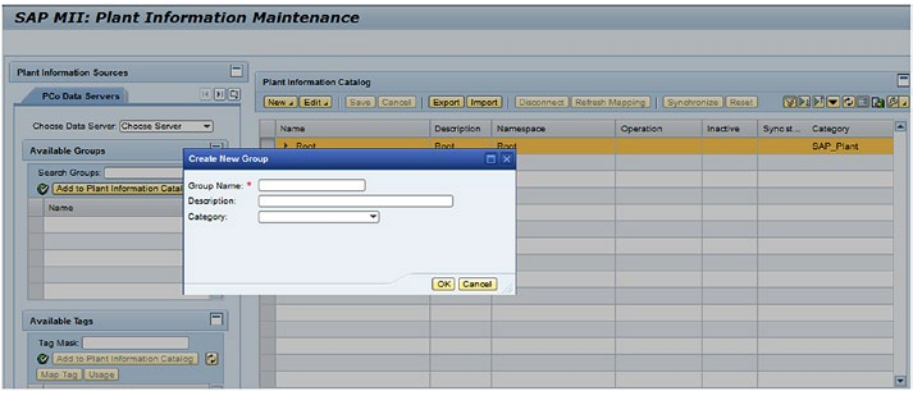


Figure 7-121. *Creating New Group PIC*

Once the object is created, there are five tabs available just below the screen with the details of that particular object:

- *General*: Specifies the general properties like name, description, namespace, and category.
- *Property Set*: Specifies the associated property sets to which values can be assigned.
- *Custom Property*: Specifies the custom properties added to that object with corresponding values.
- *Log*: Specifies the list of activities taken on that particular object.
- *Translate*: Specifies the language dependent description text for the selected object.
- *Edit*: Upon clicking this button, all edit functionalities like Edit, Cut, Copy, Paste, and Delete will be available (see Figure 7-122).

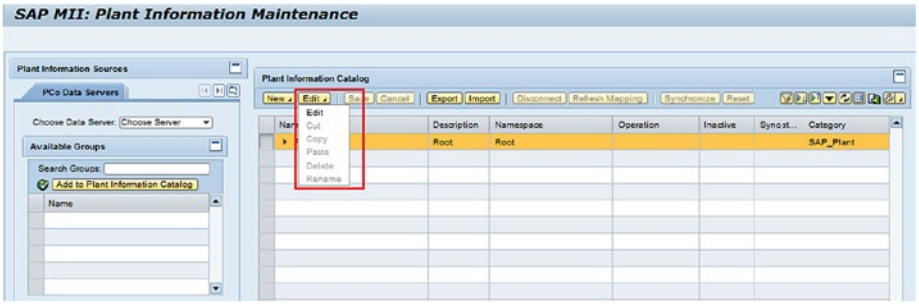


Figure 7-122. *Editing the PIC*

- *Save*: This button will be enabled when the Edit button is clicked. It allows you to save the changes made to the object.
- *Cancel*: This button will be enabled when the Edit button is clicked. It allows you to discard all the changes made after clicking on the Edit button.
- *Export*: Upon clicking this button, it exports the whole hierarchy of the selected node with all its child objects in XML format.
- *Import*: Upon clicking this button, it imports the whole hierarchy available in the XML format file with all its child nodes available in that XML file.

Adding Groups and Tags

Now we are looking at the left side of the screen, i.e., the Plant Information Sources section. This section includes the PCo Data Server tab, which lists all the shopfloor systems connected to SAP MII through PCo connectors of SAP MII. The desired data server can be selected and the proper tags can be searched and mapped or added. See Figure 7-123.

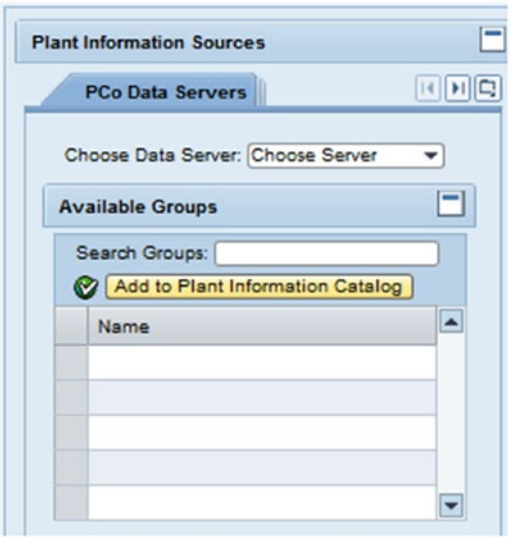


Figure 7-123. *Adding already added PCo tags to PIC from PCo data server*

There are two ways to map/add shopfloor tags to a PIC hierarchy. They can be added to the hierarchy under a group by clicking the Add to Plant Information button. Another way is to create a tag in the hierarchy and then map it to a tag from the list of shopfloor tags using the Map Tag button. The Usage button of this section shows the PIC tags mapped to that shopfloor tag in a popup.

Once a tag from the PIC hierarchy is mapped to a shopfloor tag, all the properties can be mapped to the metadata of the shopfloor tags. In that case, property values cannot be assigned manually. Values will be fetched from the shopfloor and stored. The Refresh button can be used to fetch the tag value from the shopfloor again. See Figure 7-124.

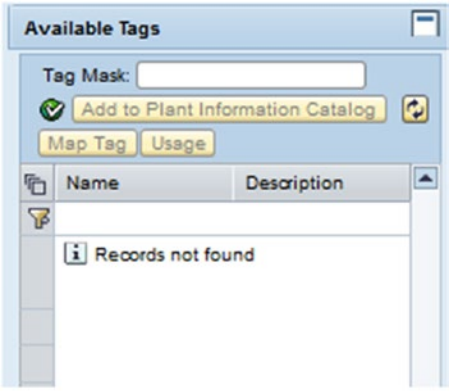


Figure 7-124. *Adding PCo tags created from the PIC and mapping them to the PIC*

Only one user can work on an object at a time. If a user is working on any object, that object is locked by his/her name shown in the Operations column of the PIC hierarchy and cannot be edited by any other user.

Adding Business Context

To add business context, ERP system data needs to be fetched. SAP MII has five configurable resource adapters to serve this purpose. These adapters allow you to bring all five resources to one point and fetch the data from there.

You configure ERP systems in the resource adapters using the following steps:

1. Log in to SAP NetWeaver Administrator.
2. Navigate to Configuration ➤ Infrastructure ➤ Application Resources.
3. Search the Plant Information Catalog. The list of all five adapters will be shown.
4. Go to the Properties tab.
5. Enter the required fields: Name, Type, Value, and Description.
6. Click Save.

Once the configuration is complete, ERP objects can be mapped to the PIC objects from the ERP Resources tab available in the Plant Information Sources section.

Approving the Change List

The hierarchy is ready now and needs to be approved. Unless the change in hierarchy is approved by an authorized person, it will not move to the PIC from the Plant Information Maintenance section. Users should have the SAP_XMII_Approver role assigned to their UME profiles to have the authorization to approve or reject the changes to the PIC hierarchy. The Catalog Services ➤ Approval List menu path will show the list of change lists submitted for approval with an option to display the details of that change list, and approve or reject them. See Figure 7-125.

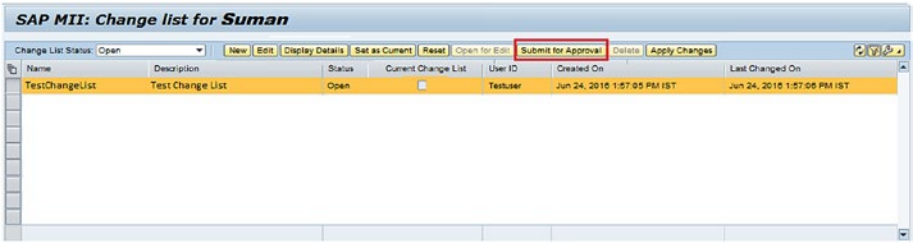


Figure 7-125. *Submitting the change list for approval*

Applying Changes to the Change List

Once the change list is approved, it comes to the same list again with the status approved and waiting for the changes to be applied. The Apply Changes button will be enabled for the change lists with approved status. Once Apply Changes is clicked, the locks acquired on the objects of that change list are released and the changes are reflected in the Plant Information Catalog screen. See Figure 7-126.

Figure 7-126. *Approval list screen*

Consuming the PIC Hierarchy

In the Query template, there is the Catalog query. This is specifically used to consume the PIC hierarchy. When a PIC is created, PIC tags are mapped to the PCo data server's tags or to the shopfloor tags. PIC saves the mapping, not the data. When a PIC tag is queried, live data is fetched from the shopfloor or PCo Data Server at that moment.

But ERP object mapping works differently. As soon as the mapping is done, all business contexts are saved to the corresponding properties in the local SAP MII system.

Follow these steps to create and configure the Catalog query:

1. Launch Workbench.
2. Go to the Catalog tab.
3. Right-click on any folder and select New ➤ Catalog Query, as shown in Figure 7-127.

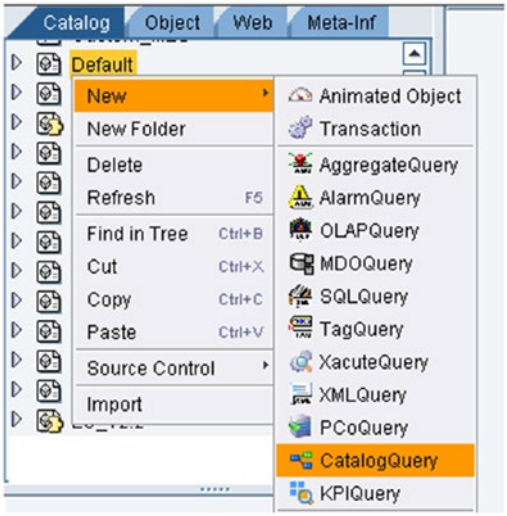


Figure 7-127. *Selecting a Catalog query*

The Template Categories options shown in Figure 7-128 are available to configure the Catalog query.

Figure 7-128. *Template Categories configuration in Catalog query*

4. From the Data Source screen, select the source data server and mode of the query (see Figure 7-129). Available modes are:

   - *ModeList*: Shows the list of modes.
   - *GroupList*: Shows the list of all groups available in the PIC hierarchy.
   - *TagList*: Shows the list of all tags available in the PIC hierarchy.
   - *Current*: Shows the current value for the tags and property values.
   - *History*: Shows interpolated historical value of the tags and the property values.
   - *HistoryEvents*: Shows the historical value of the tags and the property values.
   - *Statistics*: Shows the statistical data as per shopfloor records.
   - *RetrieveProprties*: Shows the property values.

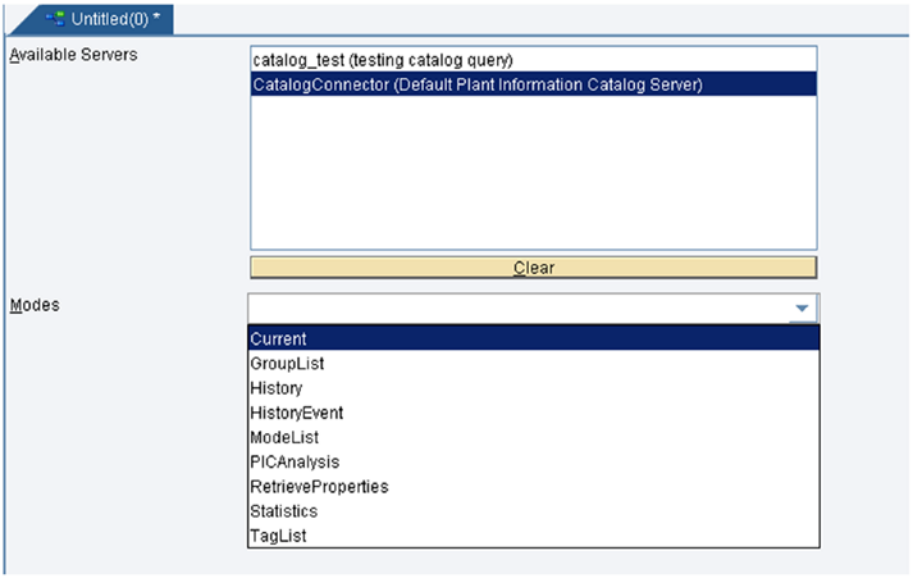


Figure 7-129. *Available modes in the Catalog query*

5. In the Plant Information Catalog Query Details screen, navigate to the PIC hierarchy of the selected data server in the PIC tree. See Figure 7-130.

6. Move the tag objects to the selected tags section either by dragging and dropping or using the available buttons.
7. Select Tag from the selected tags list and select the corresponding properties.
8. From Extended tab, SELECT the catalog server and the PCo Server used for mapping.
9. Select metadata for all the PCo servers.

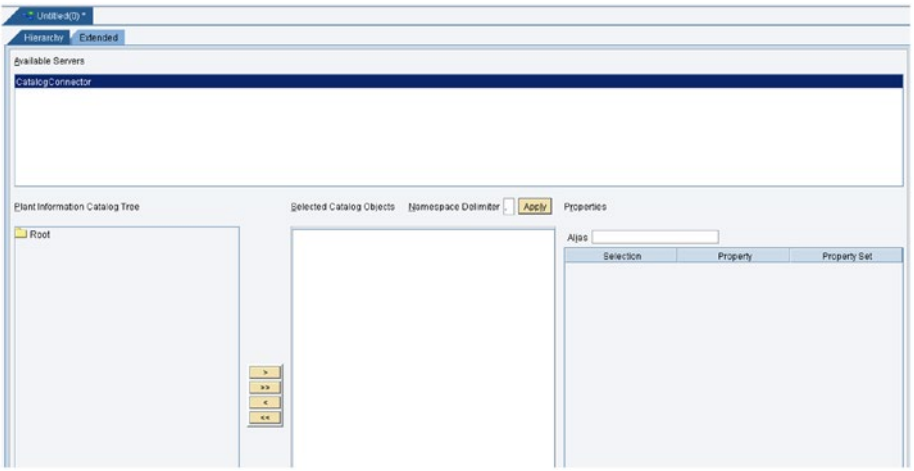


Figure 7-130. *Configuring hierarchy Catalog query*

10. The Retrieve Properties Details screen is quite similar to the Plant Information Catalog Query Details screen and is used for the Retrieve Properties mode. There is an option on this screen to choose groups as well as tags.

Catalog queries can also be used in illuminator services as shown:

```
<Protocol>://<Server IP>:<Port>/XMII/Illuminator?QueryTemplate=<Query
Path>&content-type=text/xml&Group=<Group Name>
```

PIC Web Services

PIC provides four web services. All the operations on PIC can be done using any of those web services.

The web services are as follows:

- Catalog Service
- Catalog Admin Service

- Change List Service
- Property Set Service

Catalog Service

Using Catalog Service, you can perform all the operations available using the Plant Information Catalog Screen as methods within the service. The WSDL URL for this web service is:

```
<Protocol>://<Server IP>:<Port>/PlantInformationCatalogService/PlantInformation
CatalogServiceBean?wsdl&mode=ws-policy
```

Available methods are:

- BrowseGroups
- BrowseTags
- FetchAllCustomProperties
- GetTagDetails
- GetGroupDetails
- GetCategories
- GetObjectProperties
- GetPropertyValue
- SearchGroups
- SearchTags

Catalog Admin Service

Using Catalog Admin Service, you can perform all the operations that can be performed using Plant Information Maintenance screen available as methods within the service. The WSDL URL for this web service is:

```
<Protocol>://<Server IP>:<Port>/PlantInformationMaintenanceService/Plant
InformationMaintenanceServiceBean?wsdl&mode=ws-policy
```

Available methods are:

- AddProperties
- BrowseGroups
- BrowseTags
- CreateGroup

- CreateTag
- DeleteGroup
- DeleteTag
- DeleteProperties
- GetGroupDetails
- GetTagDetails
- GetPropertyValue
- MaintainGroup
- MaintainTag
- SearchGroups
- SearchTags

Change List Service

Using Change List Service, you can perform all the operations that can be performed in Change List Screen available as methods within the service. The WSDL URL for this web service is:

```
<Protocol>://<Server IP>:<Port>/ChangeListServices/ChangeListServicesBean?
wsdl&mode=ws-policy
```

Available methods are:

- CreateChangeList
- SearchChangeLists
- ManageChangeListStatus
- ResetObjects
- ApplyChanges

Property Set Service

Using Property Set Service, you can perform all the operations that can be performed in the Property Set screen available as methods within the service. The WSDL URL for this web service is:

```
<Protocol>://<Server IP>:<Port>/PropertySetServices/PropertySetServicesBean?
wsdl&mode=ws-policy
```

Available methods are:

- AddProperties
- DeleteProperties
- CreatePropertySet
- EditPropertySet
- DeletePropertySet
- SearchPropertySets

What Is SAP Plant Connectivity (PCo)?

SAP Plant Connectivity (PCo) is a framework and set of services and management tools that provides bidirectional communication paths among control systems, control devices, files, historians, TCP sockets, and SAP. It is developed on Microsoft.Net.

Why PCo?

In any manufacturing industry, there are numerous machines and devices that continuously generate huge amounts of process data. It's critical to ensure that that data is analyzed properly and acted upon in real-time to ensure smooth operations and quick decisions in the case of deviations.

With SAP MII and PCo, real-time integration with manufacturing automation system can be easily achieved, even when the underlying systems provides different types of connection protocols. To drive real-time intelligence, PCo acts as a connection adapter and SAP MII provides the business logic to manage data and incorporate business rules.

History of PCo

PCo is the second generation of the SAP manufacturing connector technology. It replaces the xMII UDS (Universal Data Server) product that was originally developed by Lighthammer Corp. and brought into SAP with the acquisition of Lighthammer by SAP in July, 2005.

The original release of PCo, version 2.0, was in December 2008. That release only supported the real-time event-based notifications, and provided the following agents and destinations.

Agents:

- *OPC DA Agent*: *Supports OPC DA 2.05a and 3.0 specifications*
- OPC A&E Agent
- *OPC UA Agent*: Supports subscribing to OPC UA variable data, change events, and OPC UA v1.0 build 224 [and later] builds
- Socket Agent: Enables receiving streams and oriented socket messages

Destinations:

- *MII Destination*: Supports delivering notification messages to MII 11.5, 12.0, and 12.1
- *RFC Destination*: Used for enabling bi-directional communication between EWM and the socket agent and only used with the socket agent for the EWM scenario
- Does not provide general RFC usage

Integrating PCo with Automation Systems

PCo provides the following connectors as source systems:

- OPC Data Access (OPC DA)
- OPC Historical Data Access (OPC HDA)
- OPC Alarms and Events (OPC A&E)
- OPC Unified Architecture (OCP UA)
- Object Linking and Embedding Database (OLEDB)
- Open Database Connectivity (ODBC)
- iHistorian
- OSISoft PI
- TCP/IP Sockets
- File Monitor

It provides a Query mechanism using tag queries, PCo queries, and SQL queries from SAP MII and retrieves data from the underlying system on an on-demand basis.

PCo also provides Notification mechanisms. Subscription to one or more tags of the source system is possible to receive PCo notification to the target system for tag value changes and when a particular condition is satisfied.

PCo can send notifications to the following destination systems:

- SAP MII
- SAP ME (Manufacturing Execution)
- SAP HANA Database
- SQL Server
- Business Enterprise System (SAP ERP/EWM)

It is possible for PCo to send unprocessed tag information to the destination systems. But to make it more meaningful, SAP MII is used as middleware to easily develop the logic to attach corresponding machine or process parameters to the data received from PCo and consolidate it to the actual destination system for further processing.

Machine Data Connection

PCo connects directly to the automation control systems with SAP MII. First a source system in PCo needs to be configured to the automation control system. PCo can connect to SCADA or the Plant Historian System through an OPC server or directly to the source system through connectors. Certain parameter values of a machine signifies the machine downtime and are commonly known as tags. These tags are available in the Plant Data Historian. OPC simulators are generally used as automation system simulators.

Being a Microsoft Windows executable program, it is recommended to install PCo and source systems in the same Windows server to avoid complex firewall and DCOM security configurations.

What Is OPC?

OPC stands for OLE for Process Control. OPC is a software interface standard that allows Windows programs to communicate with industrial hardware devices (such as the PLC device).

OPC implementation is done in client/server pairs where the OPC server is a software program to convert hardware communication protocol used by PLC into OPC. See Figure 7-131.

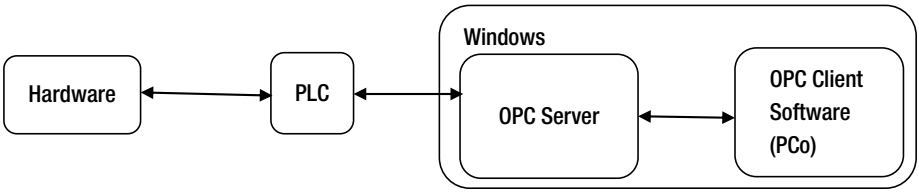


Figure 7-131. *Data flow diagram and placement of OPC*

Protocol and OPC client software is any program that needs to connect to the hardware, such as PCo. The OPC client uses the OPC server to get data from or to send commands to the hardware.

OPC is an open standard and results in lower cost for the manufacturer and more options for users. Hardware manufacturers need to provide only a single OPC server for their devices to communicate with any OPC client. Software vendors simply include OPC client capabilities in their products and they become instantly compatible with thousands of hardware devices. Users can choose any OPC client software they need, resting assured that it will communicate seamlessly with their OPC-enabled hardware and vice versa.

Configuration in PCo Management Console

Follow these steps for source system configuration:

1. Open the PCo Management console.
2. Click on New Source System icon.

3. Select Type of Source System from the drop-down (Suppose OPC DA Source System). Select the Source System Type depending on the source system connector type.
4. Specify the name of the source system.
5. Specify the description of the source system.
6. Click OK. the Source system is created now.
7. Select the newly created source system. It shows the corresponding connectivity details in the right pane.
8. Select the server details of the source system. See Figure 7-132.

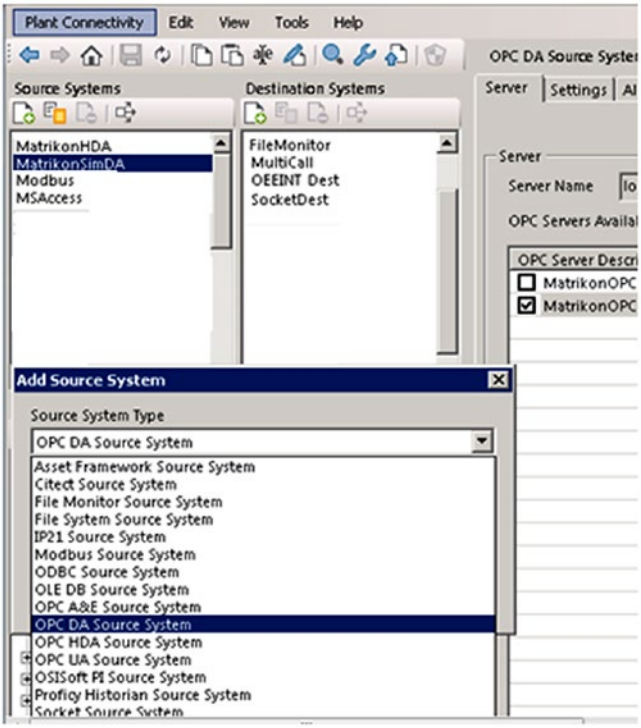


Figure 7-132. *Source System Configuration in PCo*

9. Add the server name. If the source system and PCo is installed on the same server, then the server name can be given as `localhost`. Otherwise, the hostname or the IP of the source system server needs to be mentioned.

10. PCo now will automatically determine the OPC DA server available in that particular host. See Figure 7-133.

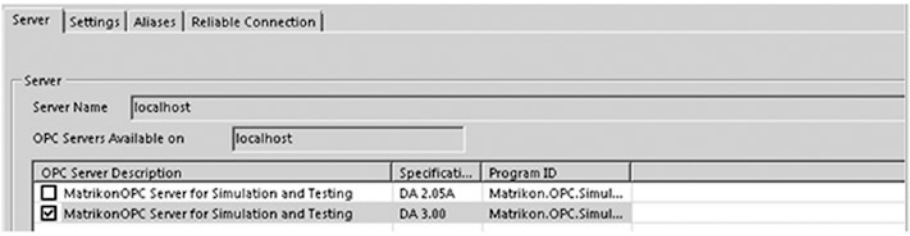


Figure 7-133. *OPC server selection in PCo*

11. Certain other properties can also be selected in the Settings, Aliases, and Reliable Connection tabs of OPC agent configuration. See Figure 7-134.

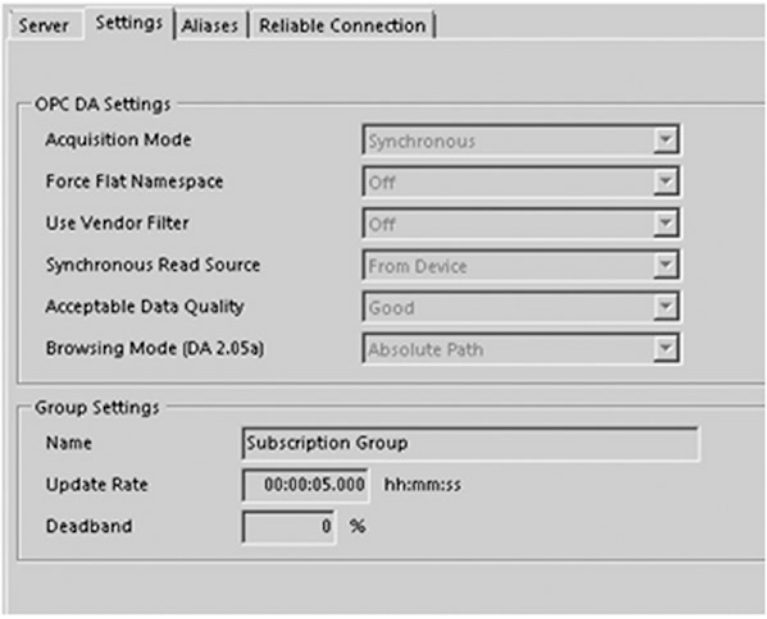


Figure 7-134. *The OPC DA setting in PCo*

The max number of retries and the retry interval can be set from the Reliable Connection window, as shown in Figure 7-135.

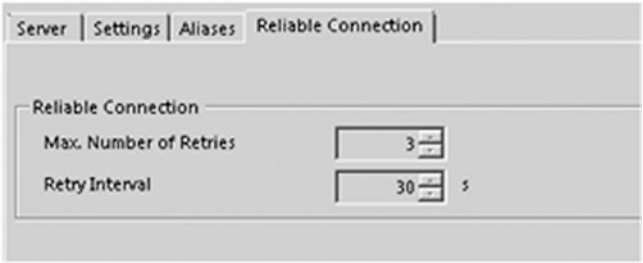


Figure 7-135. *OPC connection retention configuration in PCo*

PCo Agent Configuration

The next step is to configure the agents, the destination system, and the notification to capture real-time data. You process the data using process intelligence.

1. Click on the new icon in the agent instances from the PCo menu.
2. Select the source system for which the agent needs to be configured.
3. Specify a name.
4. Enter a description. The agent is created now.
5. Select general parameters like log level (error, fatal, information, or warning).
6. Specify a service mode and authentication if required.
7. Specify notification message queue and dispatch settings to enable queuing of notification messages sent from the PCo.
8. Create a new agent. See Figure 7-136.

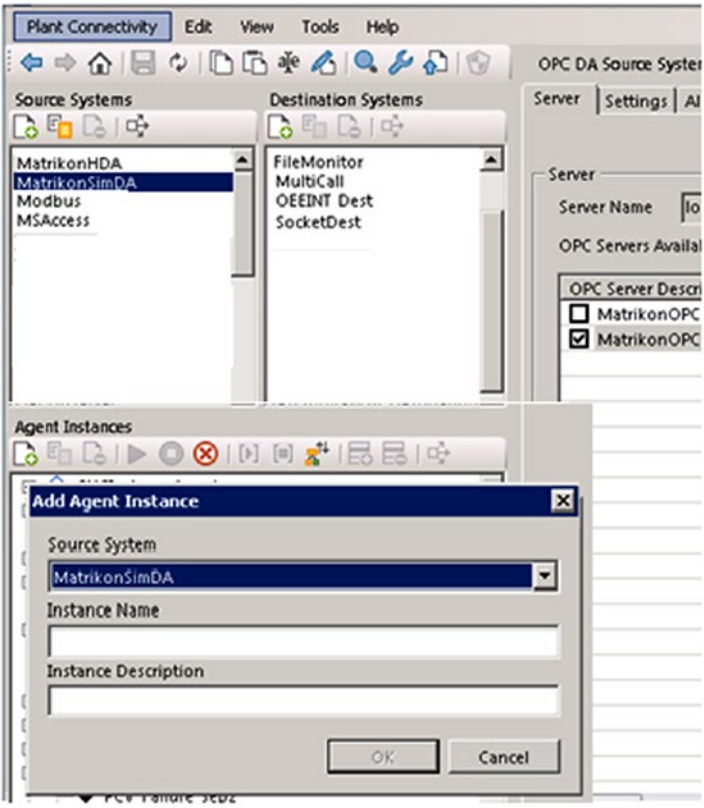


Figure 7-136. *Agent creation in PCo*

Once the new agent is created, multiple agent specific configuration needs to be done. See from Figure 7-137 to Figure 7-142.

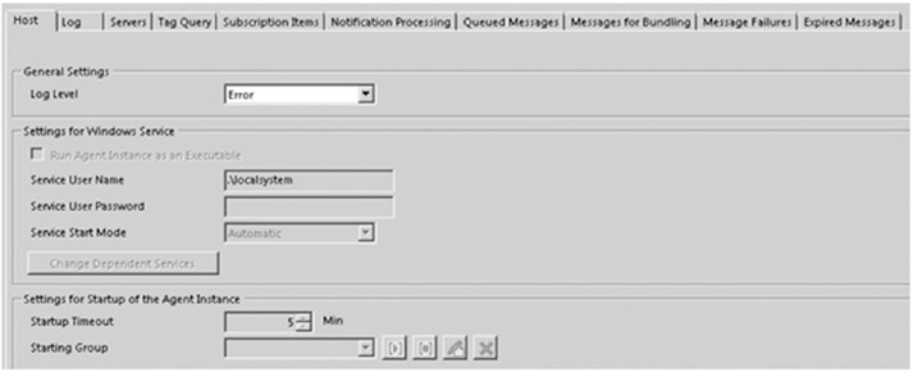


Figure 7-137. *Host configuration of created agent in PCo*

The Log tab shows all the info and error logs while reading the tag and triggering the notification. See Figure 7-138.

Figure 7-138. *The log of the agent in PCo*

The Servers tab is where you configure the port and the server type for an agent, as shown in Figure 7-139.

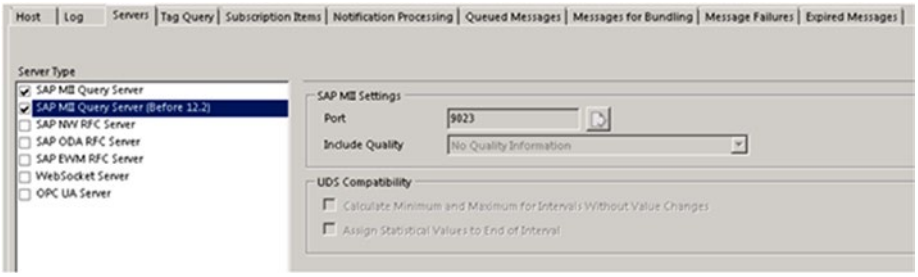


Figure 7-139. *The server type and port configuration for an agent in PCo*

The Tag Query tab is where you configure a tag query that can help to cache the data from data source, as shown in Figure 7-140.

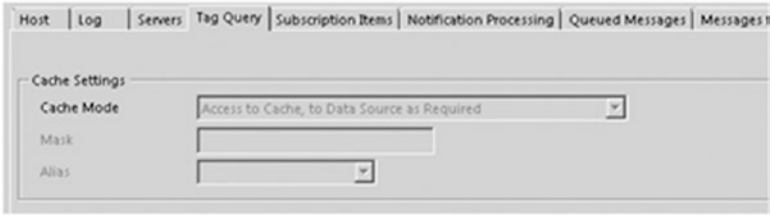


Figure 7-140. *Tag Query configuration for an agent*

The Subscription Items tab is where you add all the tags required to club together for the particular agent, as shown in Figure 7-141.

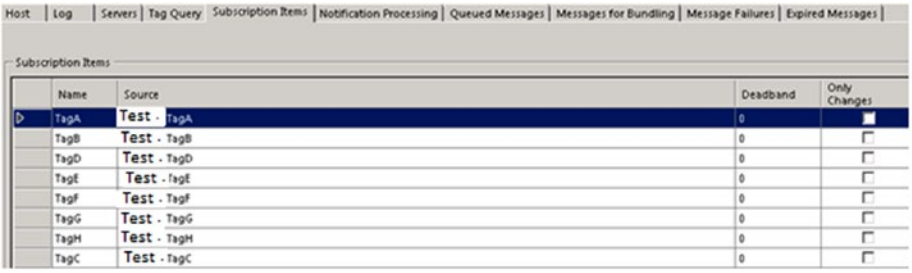


Figure 7-141. *Subscription items configuration for an agent*

9. If the application needs a notification scenario to be enabled, the corresponding tags for which the value needs to be monitored and notification needs to be triggered should be specified in the Subscription Items tab.

The Notification Processing tab is where you configure the automatic Message queuing method and dispatching rule to handle the notification, as shown in Figure 7-142.

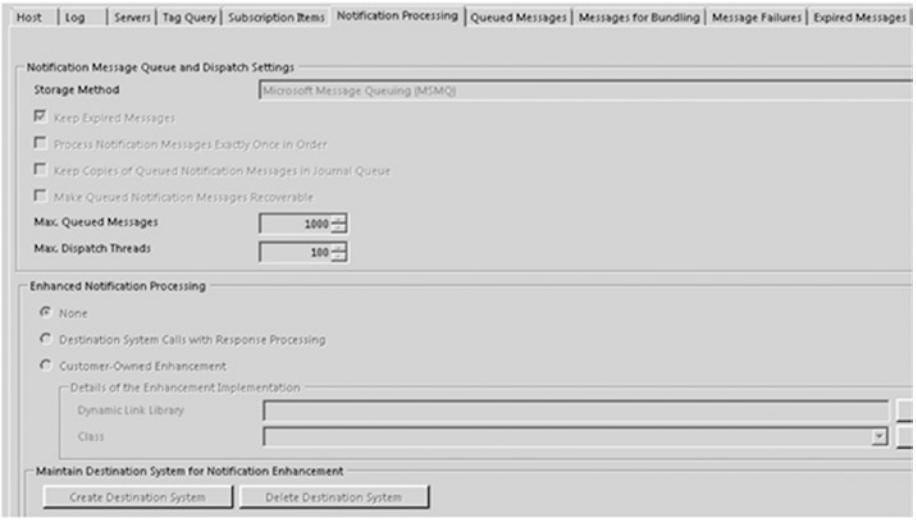


Figure 7-142. *Notification processing configuration for an agent*

Destination System Configuration

You must configure a destination system only when there is a notification scenario and PCo needs to send a notification message to the destination system for a particular change in tag values.

Destination systems are not required when tag values are queried from SAP MII on an ad-hoc basis.

Different types of destination systems can be configured in PCo to send notification messages, and one of those is SAP MII.

Once SAP MII receives notification messages from PCo, it processes the messages through business logic to add business context, as it is not possible for PCo to add all the business context. PCo messages only contain tag information.

Notification Processing Event Configuration

When a tag value in an automation system changes due to some change in a physical parameter in the machine, some action must be triggered. If a tag is maintained in the historian, a plant maintenance notification will be triggered.

For an OPC DA source system, the source system itself evaluates the condition specified in PCo for sending the signal to PCo to trigger the notification.

For any other source system, PCo evaluates the triggering condition continuously based on the tag value update rate or the event interval in the source system settings configuration to trigger the notification.

1. Select the agent created earlier.
2. Click on the Create Notification icon. See Figure 7-143.
3. Specify a name.
4. Specify a description.
5. Click OK. A notification is created.

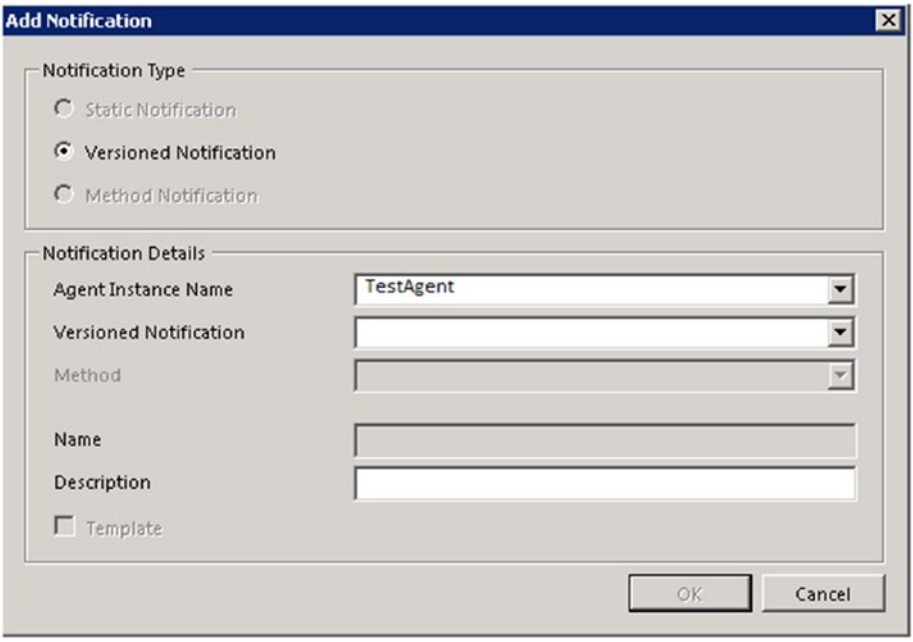


Figure 7-143. *Notification for an agent instance*

Once the notifications are added to the agent instance name, all the notification names with status and details are shown under the agent instance name, as shown in Figure 7-144.

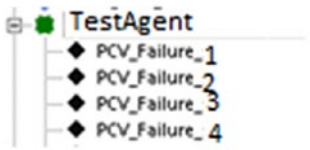


Figure 7-144. *Created notification with status*

6. Specify the trigger condition in the Notification tab. Specify any condition using the subscribed tags. , as shown in Figure 7-145.

Figure 7-145. *Trigger expression for notification*

7. Define the message format in the Output tab, as shown in Figure 7-146.

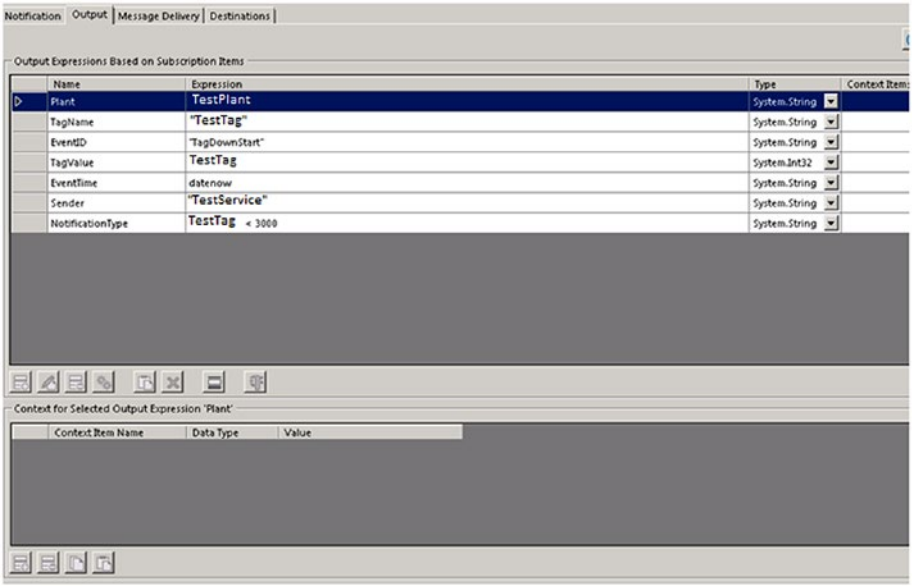


Figure 7-146. *Defining the output structure of notification*

8. Specify the conditions for message delivery in the Message Delivery tab, as shown in Figure 7-147.

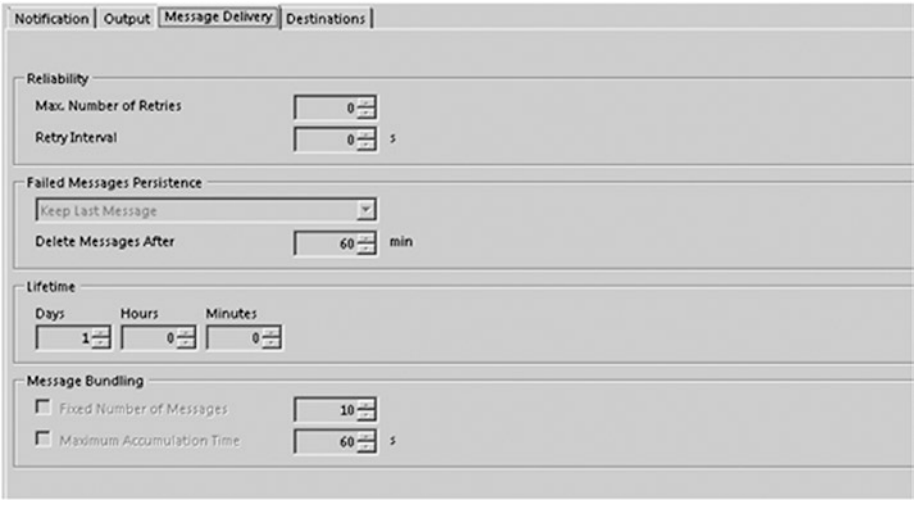


Figure 7-147. *Message delivery time configuration*

9. Specify the destination system in the Destination tab. The BLS to be executed for PCo notification processing is mentioned here, as shown in Figure 7-148.

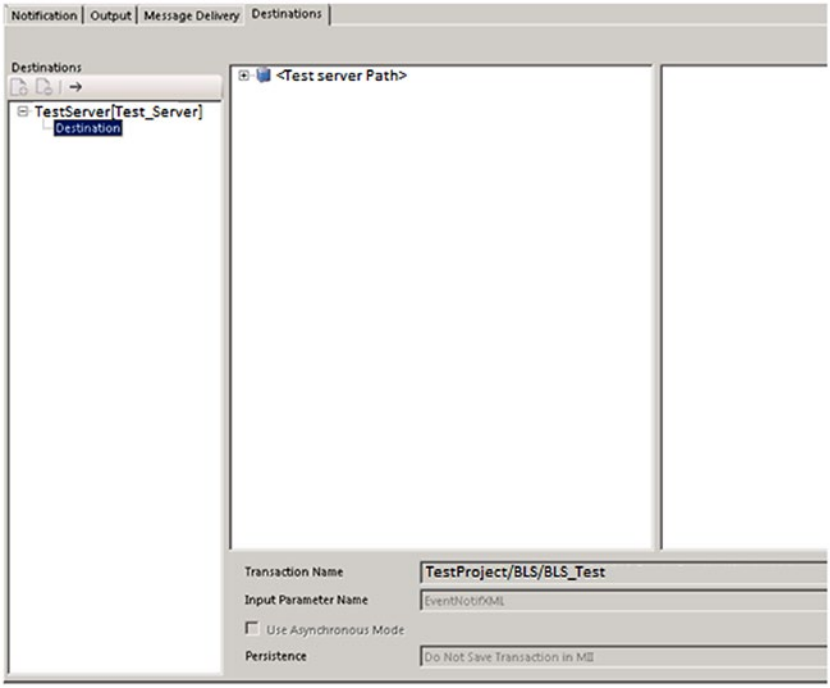


Figure 7-148. *Configuring a destination for PCo notification*

Integrating Excel with SAP MII Using SAP PCo

SAP MII can read Excel files through SAP PCo. It's become common to integrate PCo for reading Excel files (.xlsx) using SAP MII.

```
Tested Environment :
          SAP MII Version :  12.2 SP6 Patch 15
          SAP PCO : PCo 2.1.5.1
          Installation Package : Microsoft Access DB Engine X32-bit
          Connector Type : OLEDB
```

As a prerequisite, the PCo and Microsoft Access DB Engine should be installed on the same server.

■ **Note** The SAP MII version and PCo version are independent and should be configurable if the required provider exists. Reading Excel files (.xlsx) through PCo into MII requires the SAP recommended provider, called "Microsoft Office 12.0 Access Database Engine OLE DB Provider."

To read the Excel files, you need to set up some configurations in the PCo Management Console and in the SAP MII connection details. For step-by-step details, refer to this blog: https://blogs.sap.com/2015/10/26/procedure-to-integrate-pco-with-sap-mii-to-read-excelxlsx-files/.

Managing Notifications from the MII Menu Page

You can now directly manage notifications of SAP PCo from SAP 15.0 from the SAP MII menu options. You don't have to log in to the plant management console to create, configure, and manage PCo notifications. See Figure 7-149.

The benefits of having notifications managed directly in SAP MII are as follows:

- Users can manage and maintain multiple PCo installations centrally, from the same screen.
- User can directly modify, pause, and resume the notification in SAP MII.
- Enriched with the business context available in SAP MII PIC.
- Modified even when an agent instance is running, which was not possible through the Plant management console.

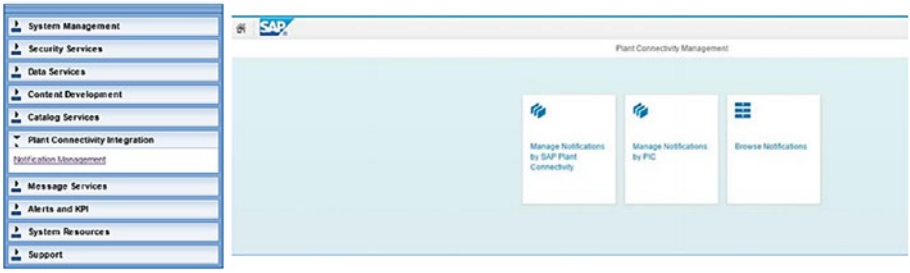


Figure 7-149. *PCo notification management from the SAP MII menu page*

As shown, when the Notification Management under Plant Connectivity Integration option is check, the Plant Connectivity Management screen opens and has the following window tabs further available:

- Manage Notifications by SAP Plant Connectivity
- Manage Notification by PIC
- Browse Notifications

Manage Notifications by SAP Plant Connectivity Screen

Once you choose Manage Notifications by SAP Plant Connectivity, you'll see the screen shown in Figure 7-150.

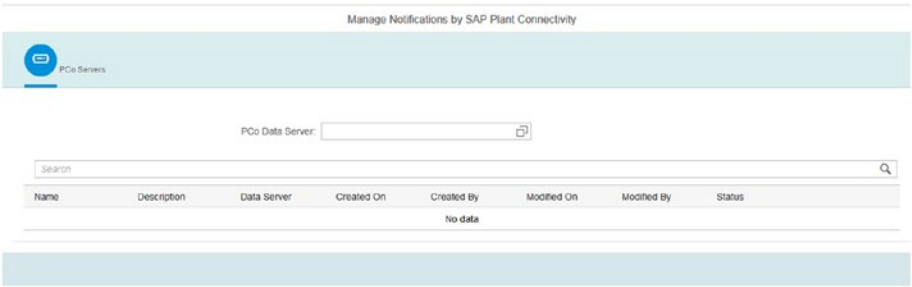


Figure 7-150. *Manage notifications by plant connectivity from the SAP MII menu page*

Once the PCo server is selected, all the notifications will be displayed in a list with descriptions and other details, as shown in Figure 7-151.

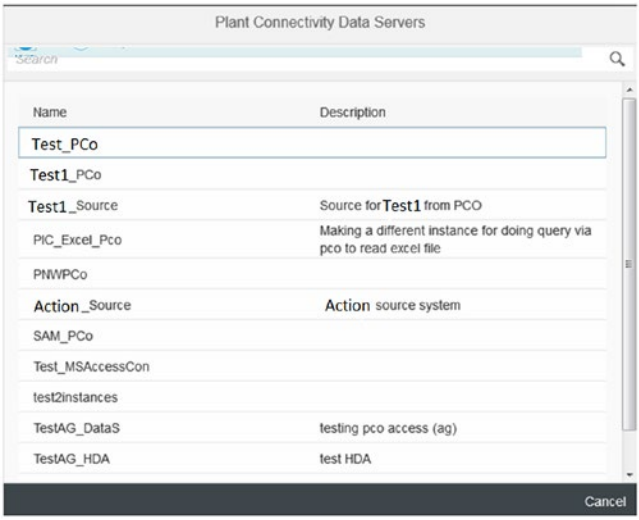


Figure 7-151. *List of notifications from the manage notification*

The PCo Servers list contains all the notifications configured for the corresponding PCo data servers configured in that SAP MII server. Users can view, monitor, and analyze against the name of the PCo data server selected by the created date, Modified On date, Modified by, and the status of the data server (running, stopped, or pending).

Once a data server is selected, the configured notifications will be displayed and if no notifications are available, then no data is displayed. In the bottom-right corner there is a button to create notifications directly in SAP MII against the data server, as shown in Figure 7-152.

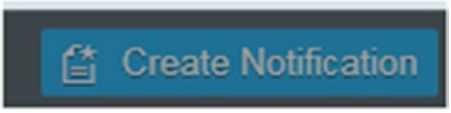


Figure 7-152. *The Create Notification button*

Manage Notification by PIC Screen

From this screen, you can view the Plant Information Catalog hierarchy and manage the notifications against the tags. If required, you can also create new notifications. See Figure 7-153.

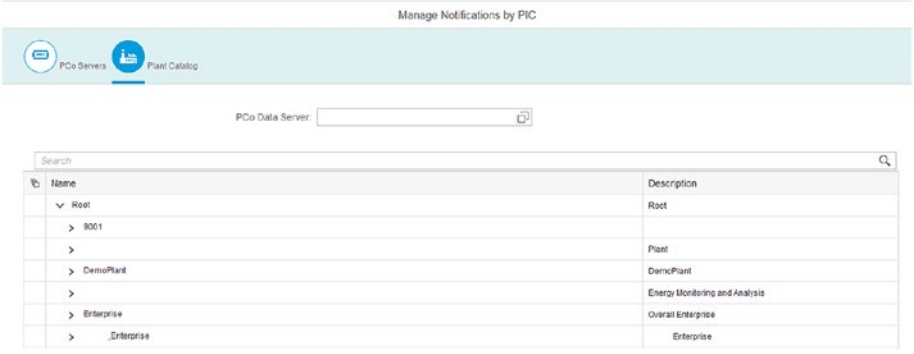


Figure 7-153. *Manage Notification By PIC screen*

Browse Notifications Screen

From this screen, all the notifications associated with the PCo data servers can be monitored for status and versioning using filter options, as shown in Figure 7-154.

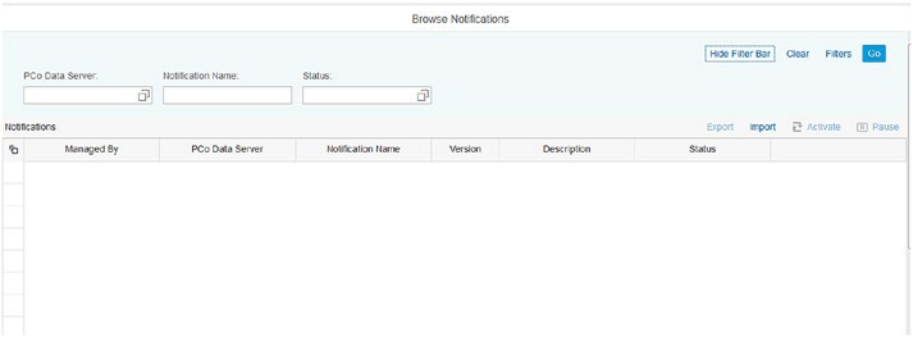


Figure 7-154. *Browse Notifications from the manage notification*

On the selected notification, you can do the following actions:

- View the notification
- Import the notification details in XML format to the SAP MII system
- Export the notification in XML format
- Activate
- Pause
- Resume

Sessioning in Queries and JRA

A *session* is a series of interactions between two communication endpoints that occur during the span of a single connection. SAP introduced some new features in MII that enable users to query SQL database logically. This ensures users can query in sessions along with using commit and rollback for appropriate transactions. The constraint is that data servers of IDBC and DataSource can use this functionality. These are the action blocks included in SAP MII Transaction for calling these specific databases in session. See Figure 7-155.

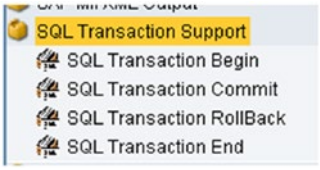


Figure 7-155. *SQL Session Support in SAP MII*

SQL Transaction Begin

This action is used mainly to begin the session for a logical database transaction. While executing, the data server is connected to the connection pool. This action block has inputs like Server Details and Connection Timeout. See Figure 7-156.

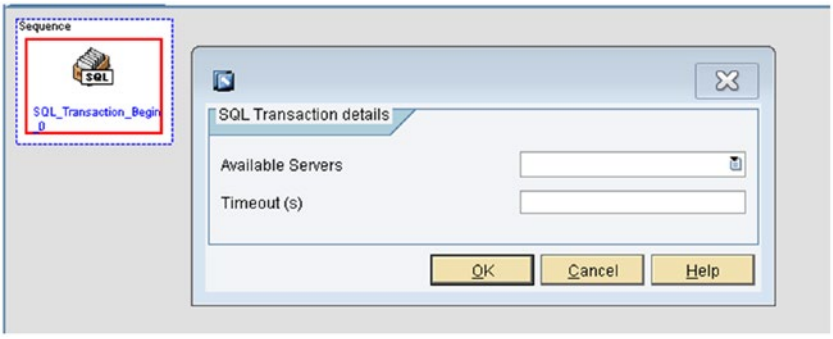


Figure 7-156. *SQL Transaction Begin action block and configuration*

SQL Transaction Commit

This action block performs commit actions on the database transaction. This action commits all the action made by the SAP MII SQL query template after beginning the transaction. The main input parameter is the SQL Transaction action block name for commit. Committing a transaction means making permanent the changes performed by the SQL statements within the transaction. See Figure 7-157.

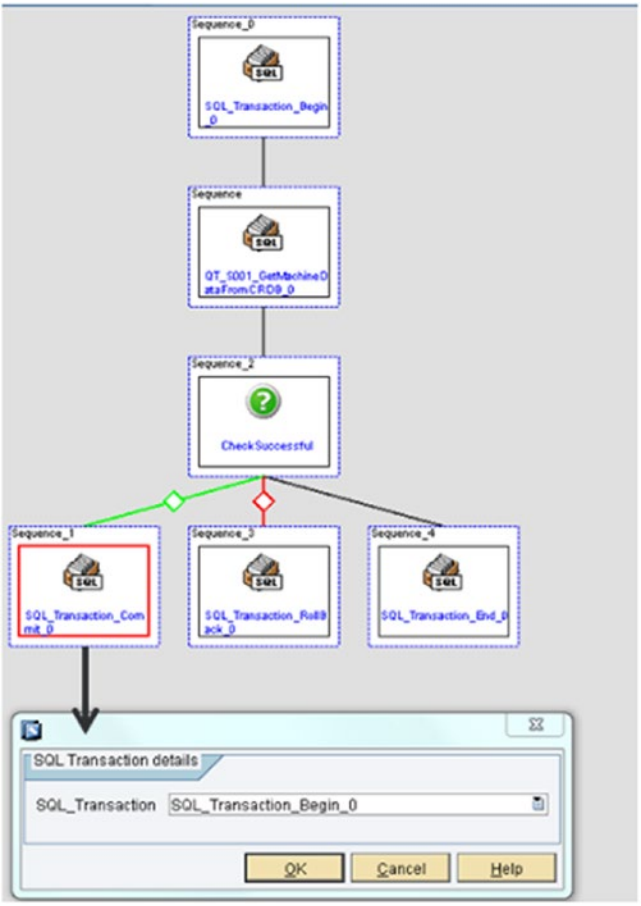


Figure 7-157. *SQL Transaction Commit action block and configuration*

SQL Transaction Rollback

This action rolls back a transaction. This option is provided to choose the transaction to roll back all the changes made in the SAP MII Transaction by the MII SQL Query Template after beginning a transaction. The main input is the SQL Transaction begin action block name. Rollback restores the state of the transaction until the last commit was performed. See Figure 7-158.

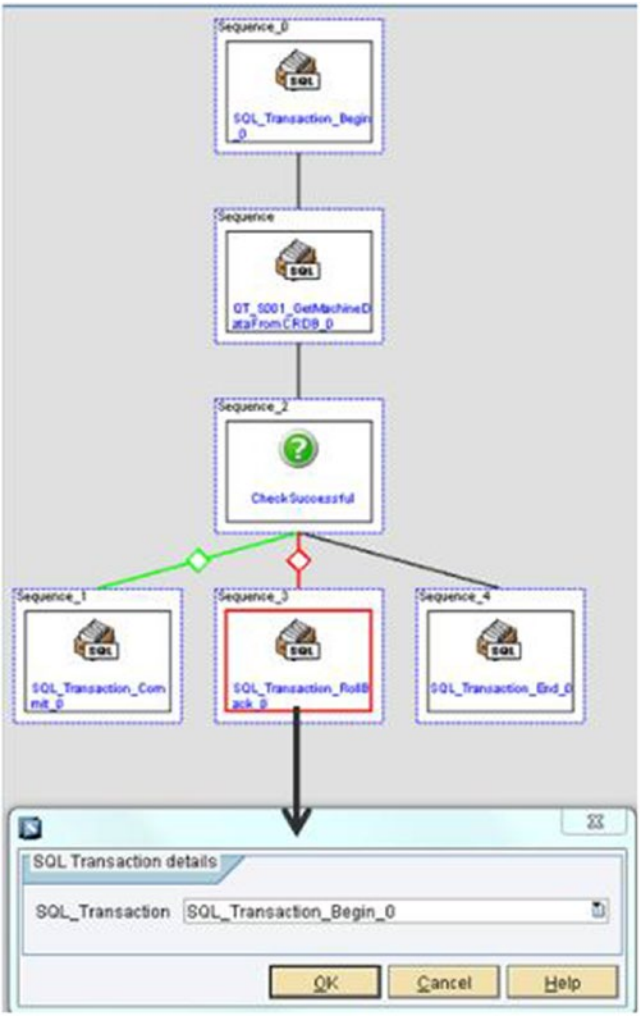


Figure 7-158. *SQL Transaction Rollback action block and configuration*

SQL Transaction End

This action ends the current database logical transaction. The input here is again the SQL Transaction begin action block name. This disconnects the data server configured in the SQL Transaction Begin action block. It is always best to end the transaction to disconnect the server when a process is completed. See Figure 7-159.

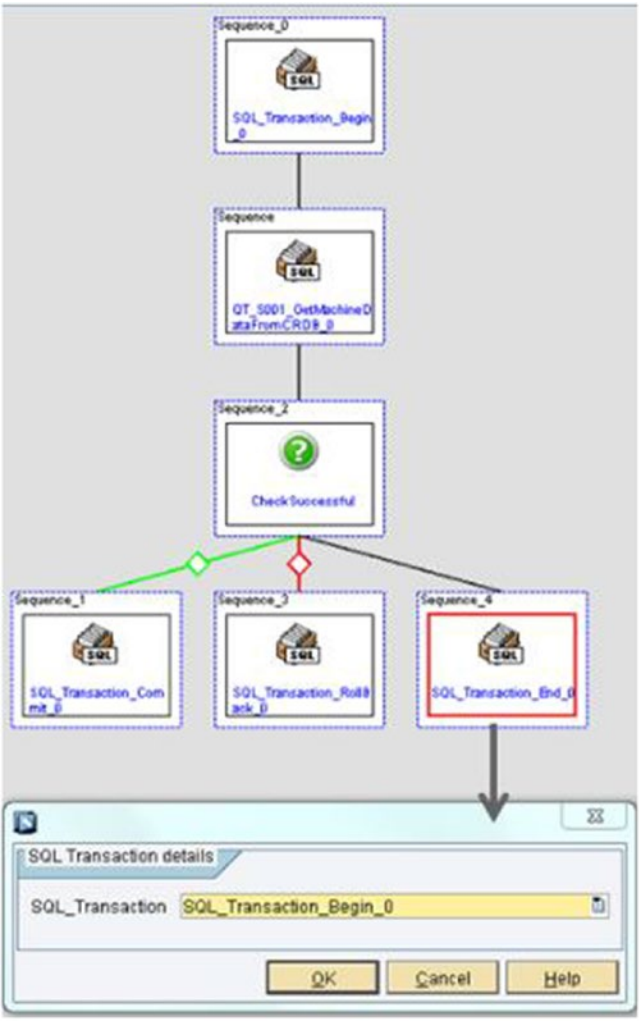


Figure 7-159. *SQL Transaction End action block and configuration*

Apart from sessions in the SAP MII SQL Query Transaction, SAP maintains SAP ECC (SAP ERP Central Component) calls from SAP MII in sessions as well. This is done for both SAP JCo (SAP Java Connector) and SAP JRA (Java Resource Adapter) connectivity.

Using SAP JRA (Java Resource Adapter)

The Java Resource Adapter (JRA) allows for both synchronous and asynchronous communication with the SAP ERP system. In SAP MII, it is used for both types of communication with SAP ERP. In this section, only the synchronous communication through action blocks is covered.

To use JRA, users need to configure the JCA connection factory in NetWeaver administrator (NWA).

Follow these steps to configure a JRA adapter:

1. Log in to the NetWeaver Administrator page (`http://<server>:<port>/nwa`).
2. Select Configuration Management ➤ Infrastructure ➤ Application Resources.
3. Select JCA Resources.
4. Select SAPJavaResourceAdapter15 from the resource list.
5. In the Resource Details section, select the Dependent JCA Connection Factories tab.
6. Choose Copy and Add New JCA Connection Factory to define the JRA connection. See Figure 7-160.

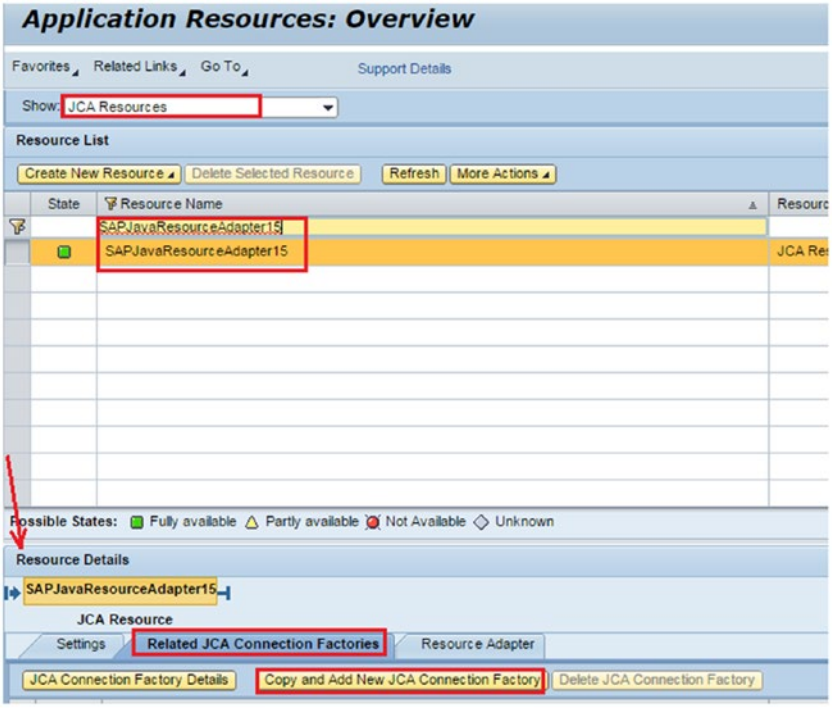


Figure 7-160. *JCA Connection factory configuration creation*

7. Enter a relevant resource name for the JNDI (Java Naming Directory Interface) name (see Figure 7-161):

   - *NonGlobal*: SAP ERP specific connections. This setting will not be enlisted into distributed (Java Transaction API or JTA) transactions.
   - *Shareable*: This default JRA operation and, as it's sharable, only one physical connection is created even if an application wants to acquire multiple connections.
   - *Unsharable*: Each connection for the application will use separate physical connections from the connection pool.

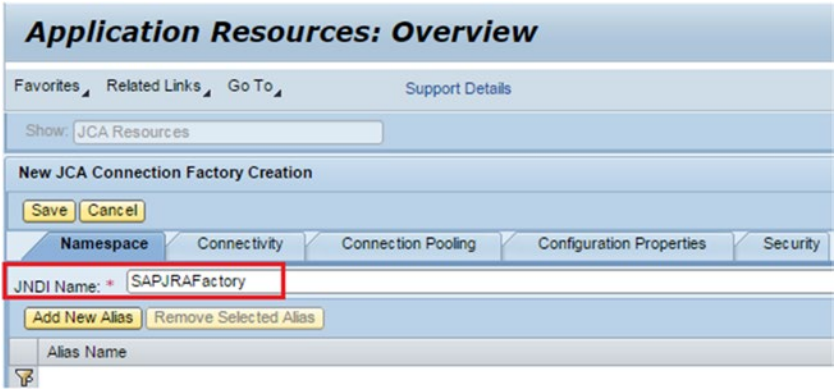


Figure 7-161. *JNDI configuration for JRA factory*

8. Go to the Configuration Properties tab.
9. Enter the following values for the JRA communication configuration (see Figure 7-162):

   - *SAPClient*: SAP ERP client number
   - *UserName*: SAP ERP client username
   - *Password*: SAP ERP password to connect to the client
   - *Language*: Enter login language for the user
   - *ServerName*: The SAP ERP server URL as it resolves on your network (fully qualified)
   - *PortNumber*: ERP defined system number you are connecting to

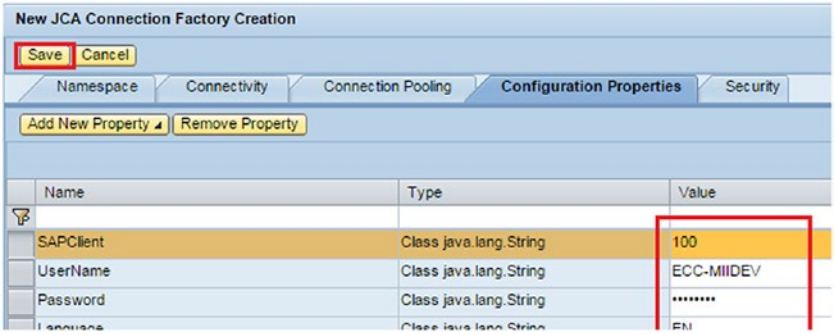


Figure 7-162. *JCA Adapter properties configuration*

10. Save the configuration now.

Now, as the JRA adapter has been configured and saved, the JRA action blocks can be used for development in SAP MII.

For SAP JRA the different action blocks in SAP MII are described here:

- *SAP JRA Start Session*: This action is to open a session between SAP MII and the SAP ERP Central Component (SAP ECC) using the SAP Java Resource Adapter (SAP JRA). Users have to select the configured JCA factory connection from the JRA Connector Name list, as shown in Figure 7-163.

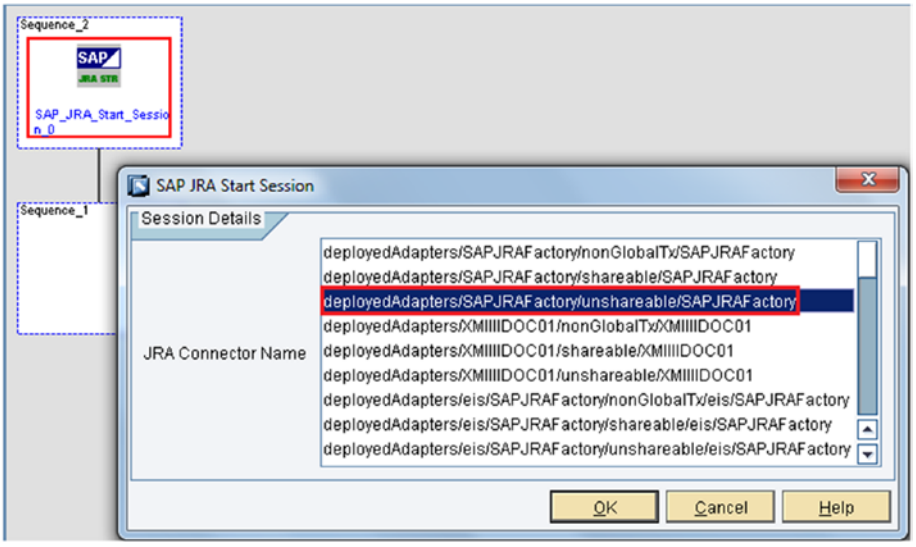


Figure 7-163. *JRA factory connector list for start session action block*

- *SAP JRA Function Call*: This is the action to call RFC. It is different from the SAP JCo function action block (which is explained later), as JRA is more secure and follows a user-specific encryption algorithm. See Figure 7-164.

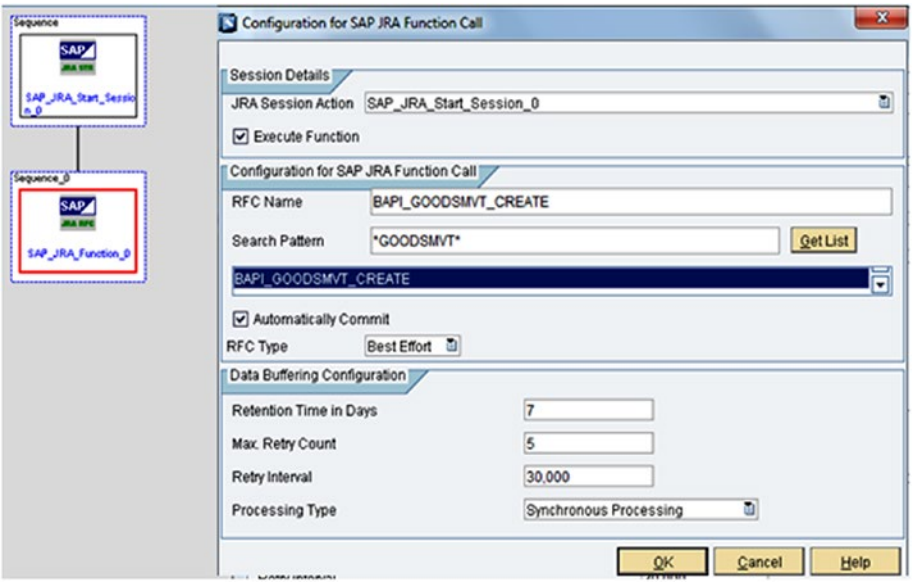


Figure 7-164. *JRA function action block configuration*

- *JRA Session Action*: This provides the SAP JRA Start session action block name.

  (Other configuration options in the JRA function block have the same functionality and will be explained later in the JCo function action block section).

- *SAP JRA Commit*: This action performs a commit function during a session between SAP MII and SAP ERP Central Component (SAP ECC) using the SAP Java Resource Adapter (SAP JRA). See Figure 7-165.

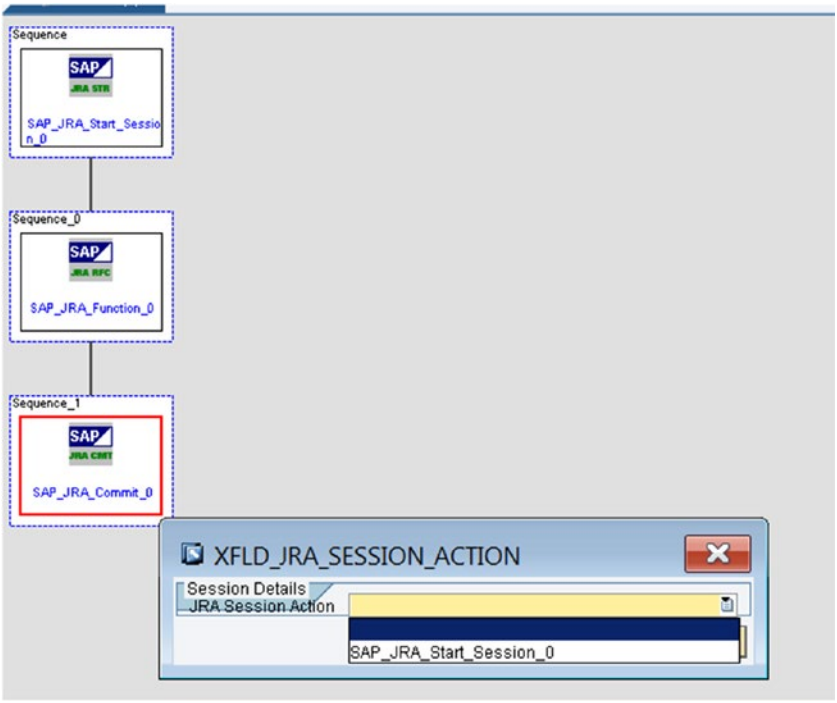


Figure 7-165. *JRA commit action block and its configuration*

- *SAP JRA Rollback*: This action can be used to roll back a function call during a session between SAP MII and SAP ERP Central Component (SAP ECC) using SAP Java Resource Adapter (SAP JRA). See Figure 7-166.

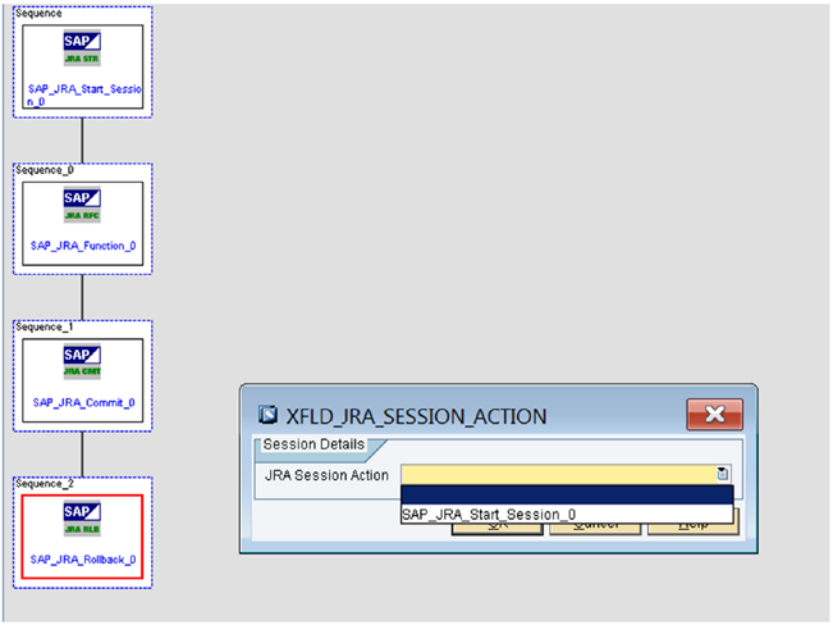


Figure 7-166. *JRA rollback action block and its configuration*

- *SAP JRA End Session*: This action can be used to close a session between SAP MII and SAP ERP Central Component (SAP ECC) using the SAP Java Resource Adapter (SAP JRA). See Figure 7-167.

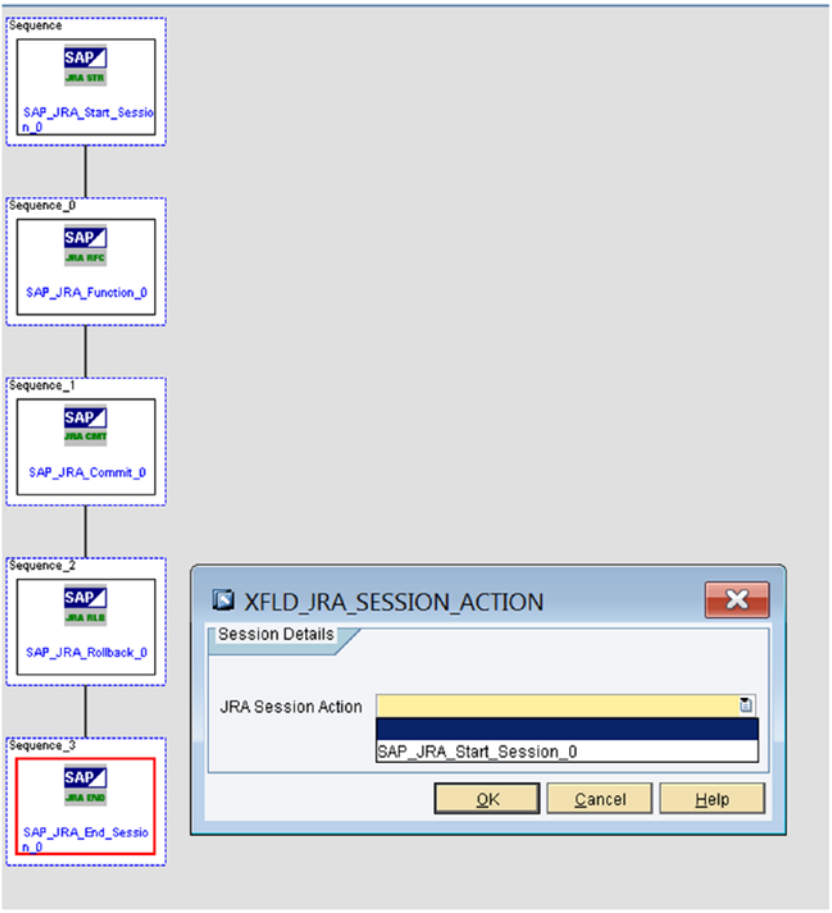


Figure 7-167. *JRA end session action and its configuration*

Using SAP JCo (SAP Java Connector)

SAP JCo action blocks call a Business Application Programming Interface (BAPI) using the SAP Java Connector (SAP JCo). They also control sessions between SAP MII and SAP ERP Central Component (SAP ECC).

The components are described here:

- *SAP JCo Interface*: This action creates a connection to the SAP Java Connector (JCo) interface from the ERP server and creates a Remote Function Call (RFC) request. The SAP JCo Interface action is used to send XML messages to and from the ERP system. This does not require session handling. See Figure 7-168.

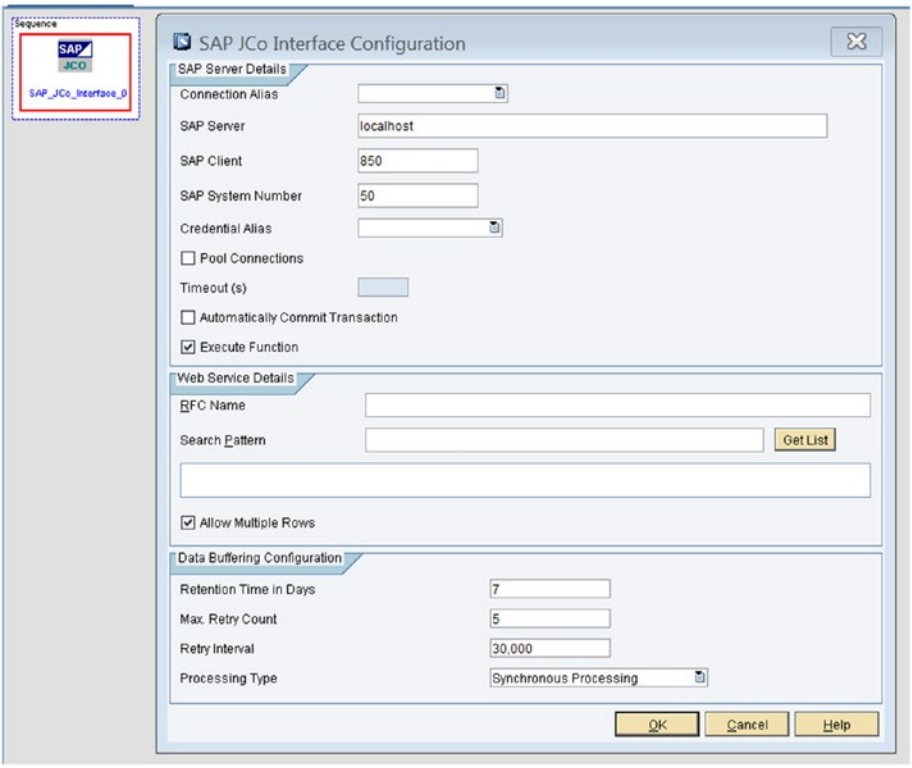


Figure 7-168. *JCo interface action block and its configuration*

- *SAP JCo Start Session*: This action opens a session between SAP MII and SAP ERP Central Component (SAP ECC). See Figure 7-169.

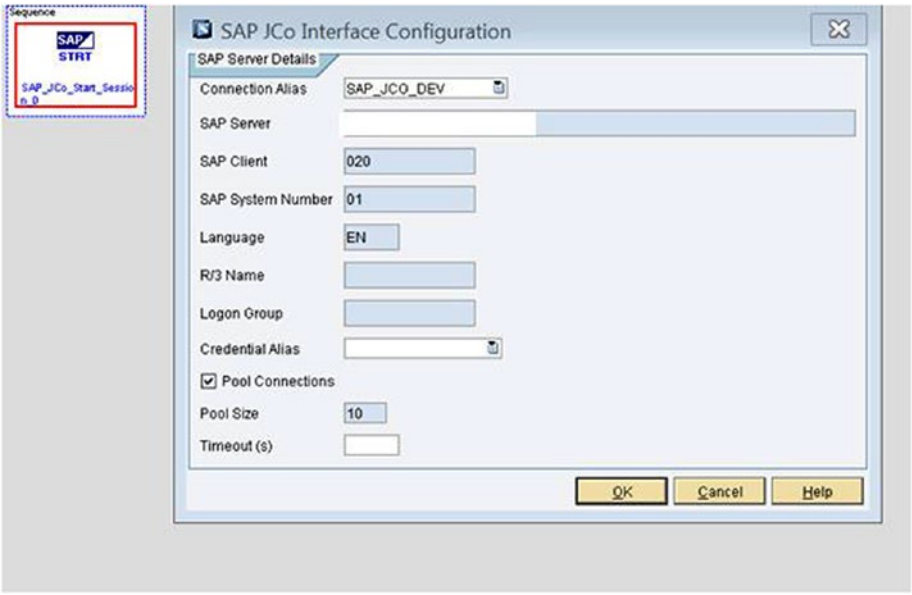


Figure 7-169. *JCo Start Session action block and its configuration*

- *SAP JCo End Session*: This action closes a session between SAP MII and SAP ERP Central Component (SAP ECC). See Figure 7-170.

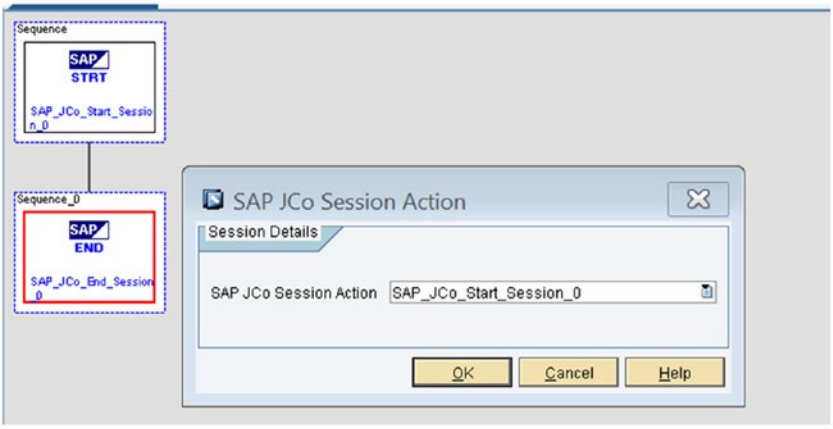


Figure 7-170. *SAP JCo End Session action block and its configuration*

This action is configured using the same Start Session action block name and is placed in the reference as shown to terminate the JCo session.

- *SAP JCo Function*: This action performs a function during a session between SAP MII and SAP ERP Central Component (SAP ECC). See Figure 7-171.

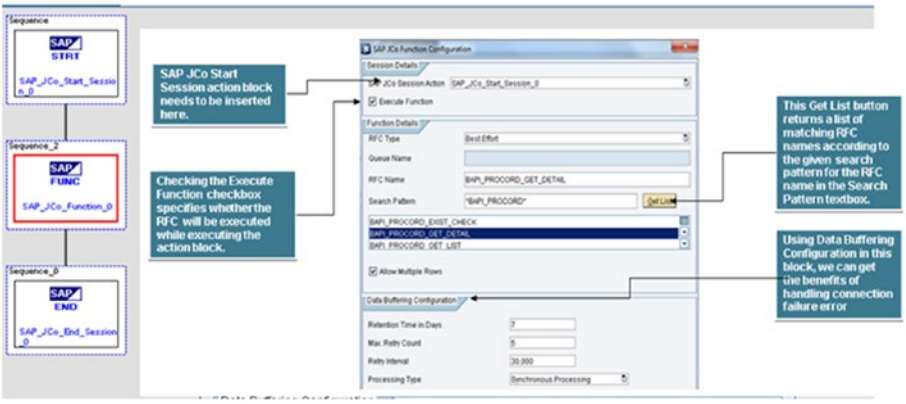


Figure 7-171. *SAP JCo function action block and its configuration*

The following often used options are there to configure the JCo function: The following RFC types are available:

- *Best Effort* (sRFC or 0)
- *Exactly Once* (tRFC or 1)
- *Exactly Once in Order* (qRFC or 2)

- *Queue Name*: The tRFC queue name used for sequential BAPI/RFC processing.
- *RFC Name*: The name of the BAPI/RFC to call.
- *Search Pattern*: Used to search for a BAPI/RFC by name. It can be used with the asterisk character for wildcard searches.
- *Retention Time in Days*: Number of days to asynchronously retry to call SAP ECC. It only applies to asynchronous processing types.
- *Max. Retry Count*: Maximum number of times to retry the call to SAP ECC. It only applies to asynchronous processing types.

- *Retry Interval*: Amount of time in seconds between SAP ECC communication retries.
- *Processing Type*: One of the following strategies used to communicate with SAP ECC:
    1. *Synchronous processing*: The BAPI/RFC is executed only once, even if it fails.
    2. *Asynchronous processing on error*: The BAPI/RFC is executed once. The transaction waits until the initial call succeeds or fails and then continues. If communication to SAP ECC is down, it retries in the background.
    3. *Asynchronous processing*: The BAPI/RFC is always executed in the background. The transaction continues without waiting for a response.
- *SAP JCo Commit*: This action performs a commit function during a session between SAP MII and SAP ERP Central Component (SAP ECC). See Figure 7-172.

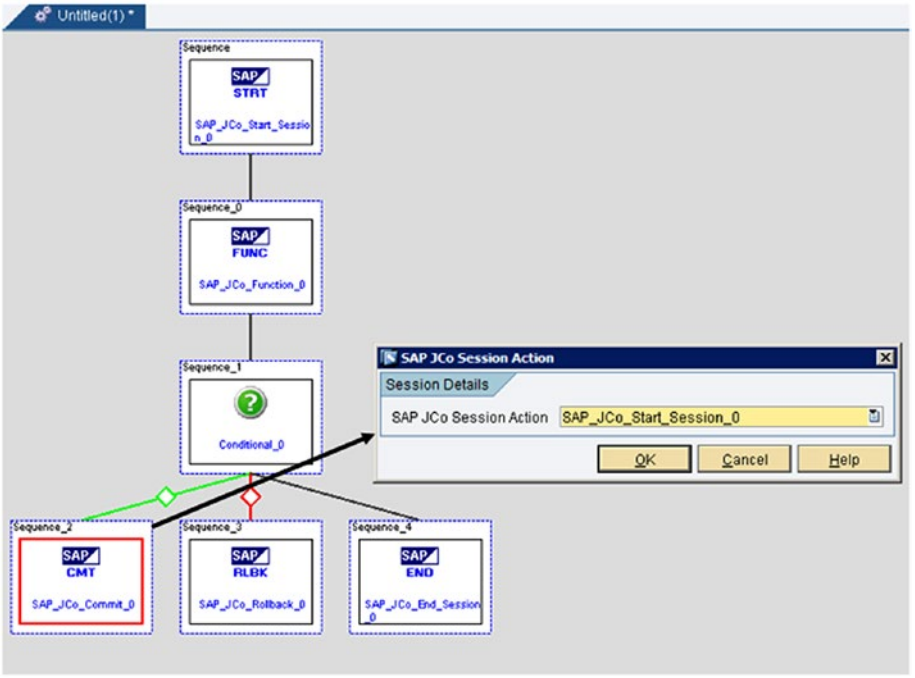


Figure 7-172. *SAP JCo commit action block and its configuration*

This action needs the JCo Start Session action block name to be referred in the JCo commit action block to commit the function.

- *SAP JCo Rollback:* This action is to roll back a function during a session between SAP MII and SAP ERP Central Component (SAP ECC). See Figure 7-173.

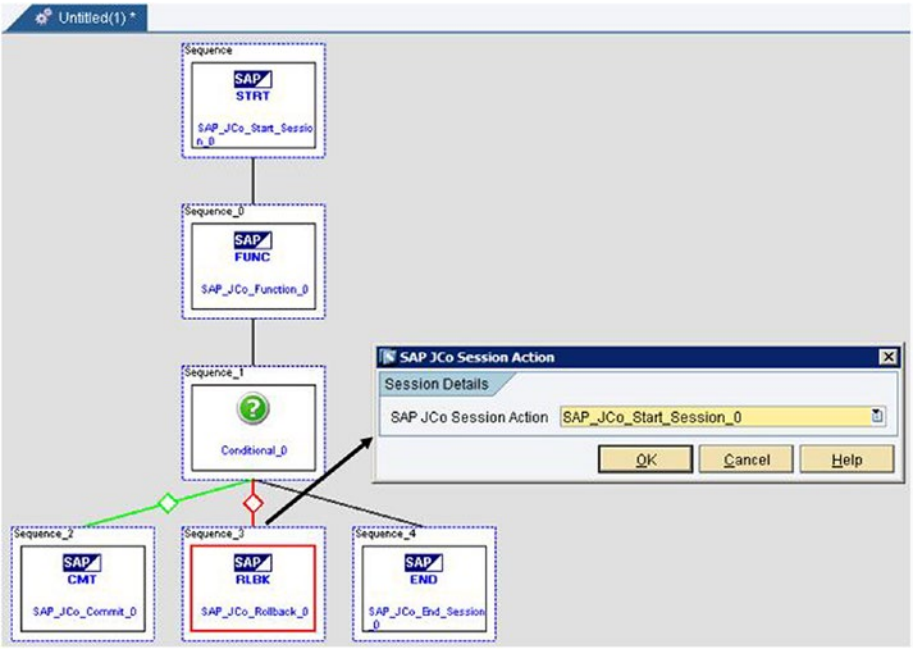


Figure 7-173. *SAP JCo Rollback action block and its configuration*

This action needs the JCo Start Session action block name to be referred in the JCo Commit action block to commit the function.

- *SAP JCo Execute Queue*: This action performs the queued Remote Function Call (qRFC) processing. Suppose a SAP JCo Function action with an RFC can be configured. The value is Exactly Once in Order and the queue name is typed. Then, SAP JCo Execute Queue action to the transaction to run the queue on the target SAP server can be added.

This action block is used with the SAP JCo Function action block whenever any queued Remote Function Calls (RFCs) are required. To achieve this, the SAP JCo Function action block needs to be configured with the RFC Type Exactly Once In Order and specified with the queue name, as shown in Figure 7-174.

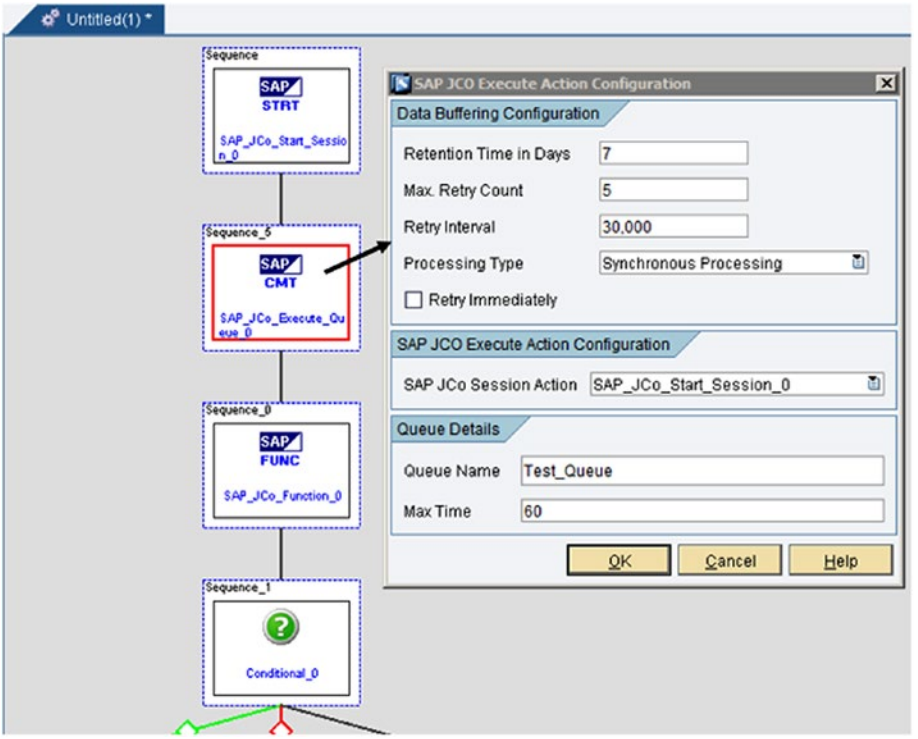


Figure 7-174. *SAP JCo Execute Queue action block and its configuration*

The system checks for an existing data buffer entry. If an entry exists, the system creates a new data buffer entry for the qRFC and adds it to the job.

New Action Blocks in SAP MII 15.X

Many new action blocks were introduced in the new versions of SAP MII. The new action blocks are described here:

- *SQL Transaction Execute Batch*: This action block is used to perform batch processing, as shown in Figure 7-175.

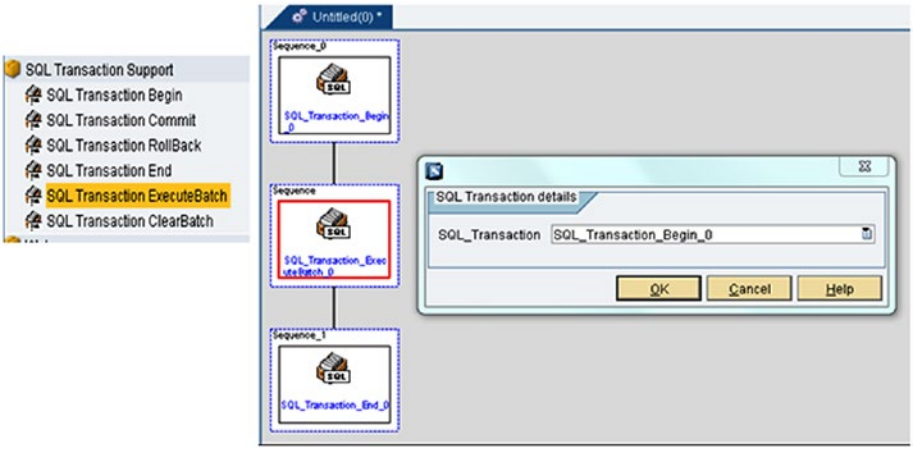


Figure 7-175. *SQL Transaction Execute Batch action block and its configuration*

- *SQL Transaction Clear Batch*: This action block clears the batch processing in the queue, as shown in Figure 7-176.

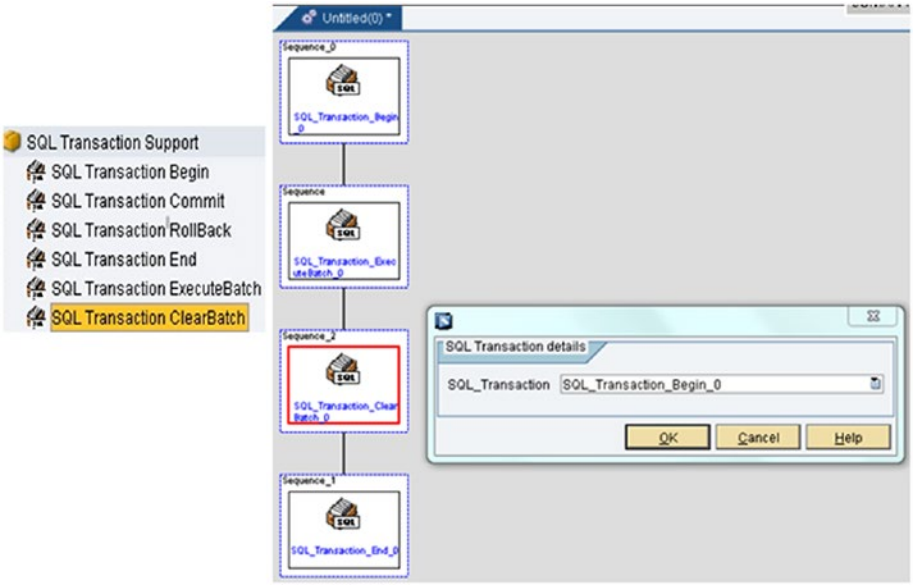


Figure 7-176. *SQL Transaction Clear Batch action block and its configuration*

- *JSON to XML Converter*: This XML function block converts the JSON structure into an XML structure, as shown in Figure 7-177.

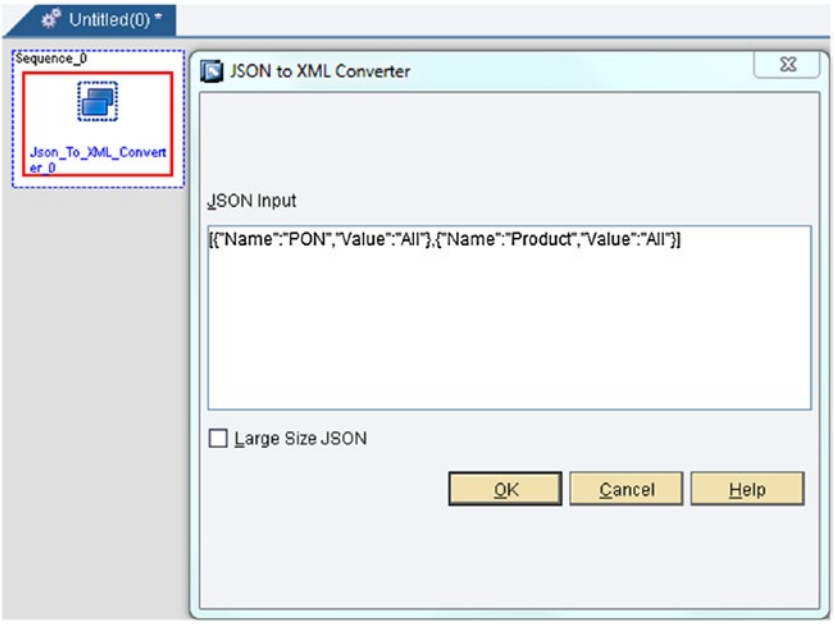


Figure 7-177. *SAP JSON to XML Converter action block and its configuration*

When an input JSON structure is provided in the JSON input and saved and executed, an XML version as an output of the action block is generated. The Large Size JSON checkbox is a Boolean flag and takes 1 or 0 when checked and unchecked, respectively. If it's selected, the large JSON Input could be provided; otherwise, it won't accept.

- *ADS Printing*: ADS printing lets you generate the file in PDF format with barcode/QR code from SAP MII. This can be directly used to print the result to a real printer, to save it in the file system, or feed it to a web browser, as shown in Figure 7-178.

  Refer to `http://www.sap.com/documents/2016/09/864b7d45-8a7c-0010-82c7-eda71af511fa.html`.

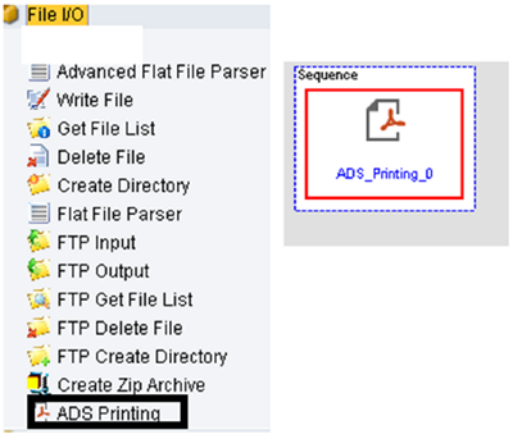


Figure 7-178. *ADS Printing action block and its configuration*

Summary

This chapter introduced the general concepts associated with MDO, visualization services, UI5 and SSCE, KPIs and alerts, PIC, PCo, session handling for queries and JRA, and the new action blocks introduced in SAP MII.

The next chapter discusses the best practices that should be followed for SAP MII and the frequently asked questions that will help you gain a more in-depth understanding of SAP MII.