SAP PRESS
First-hand knowledge.
Rheinwerk Publishing

SAP PRESS
Implementing SAP® MII
› Configure SAP Manufacturing Integration and Intelligence with step-by-step instructions
› Perform business logic service transactions, use visualization services, and set up the plant information catalog
› Master overall equipment effectiveness and improve manufacturing KPIs
2nd edition, updated and expanded
Saha · Jash · Mukherjee
Malakar · Bhattacharjee
Rheinwerk Publishing
Implementing SAP® MII
2342

Reading Sample

This sample chapter details configuring SAP Plant Connectivity to connect to supervisory control and data acquisition, data historian, and legacy plant automation systems as well as configuring the data server to connect external databases. The chapter also introduces message services, walking through the different settings, creating processing rules and cleanup rules, reprocessing failed messages, and handling successfully processed messages. It also explains how message services are used for SAP S/4HANA and MES integration with SAP MII.

"Managing Data from External Systems"

Contents

Index

The Authors

Saha, Jash, Mukherjee, Malakar, Bhattacharjee

Implementing SAP MII

705 pages | 01/2023 | $99.95 | ISBN 978-1-4932-2342-8

www.sap-press.com/5610

Chapter 3
Managing Data from External Systems

This chapter explains the integration aspects of SAP MII, including how to connect to external manufacturing plant floor systems using SAP Plant Connectivity and data servers, and how to synchronize the plant floor with other parts of the enterprise using message services.

In the previous chapter, we discussed the installation and configuration activities of SAP MII, along with server administration and required user management in SAP MII. You also learned about content management and transport using SAP NetWeaver Development Infrastructure (NWDI). In this chapter, we'll discuss required configuration in SAP MII to connect to different external data sources, such as SQL databases, LIMS, MES, SCADA, historians, and other systems to query and update data for manufacturing.

This chapter details the configurations in SAP Plant Connectivity to connect to SCADA/historians and legacy plant automation systems and the data server configurations to connect external databases. The chapter also introduces message services, walking through the different settings, creating processing rules and cleanup rules, reprocessing failed messages, and handling no rule and successfully processed messages. It explains how they're used for SAP S/4HANA and MES integration with SAP MII.

3.1 Using SAP Plant Connectivity

SAP Plant Connectivity is an adaptor component which is used to provide connectors for various types of shop floor automation systems, such as plant data historians, supervisory control and data acquisition (SCADA), distributed control system (DCS), open platform communications (OPC) servers, programmable logic controllers (PLCs), machines, and so on. There are both standard adaptors such as OPC, OLEDB, File, Socket, and Modbus, as well as proprietary adaptors such as IP21, OSIPI, iHistorian, Citect, and more to connect to specific manufacturing automation and operation management systems as well as directly to machines, devices, and control systems. SAP Plant Connectivity is usually used with SAP MII to integrate with shop floor automation systems to bidirectionally exchange data, but it can also be used directly with other systems such as SAP S/4HANA, SAP Business Technology Platform (SAP BTP), SAP Event Stream Processor, and SAP Extended Warehouse Management (SAP EWM) to integrate those with the shop floor automation systems and machines. Based on operational data generated from machines, SAP Plant Connectivity can trigger a notification for the destination system and can perform particular tasks. SAP Plant Connectivity also provides enhanced notification functionality where the source system is expecting immediate acknowledgement from SAP Plant Connectivity without involving the destination system.

SAP Plant Connectivity provides different types of connectors to connect to different types of systems and devices. It can connect with equipment and control systems (PLCs) via Socket or Modbus connectors. For equipment and devices providing data directly, such as weighing scales, typically a convertor such as an RS232 TCP/IP ethernet convertor may need to be used to convert a serial connection to ethernet by providing streaming data through the TCP/IP socket. For equipment and devices connected with a control system (PLC) Modbus connector, SAP Plant Connectivity may be used to bidirectionally exchange the data through a Socket connector. Most of the connectors available in SAP Plant Connectivity provide query and notification support, by which they can query the data from the source from SAP MII or can get a notification message in SAP Plant Connectivity (which can be sent to any other destination system) when there is a change in value in the source system or a formula/expression is satisfied based on the data in the source system. SAP Plant Connectivity integrates with equipment and devices via source system and agent configurations created in it, based on the connector types used. Ultimately, SAP Plant Connectivity needs a system to which the data can be sent or can be queried from. Now, let's discuss source systems, destination systems, agents, and notifications in detail in the subsequent sections.

Example

In this chapter, we're considering a scenario where one piece of shop floor equipment is generating quality-related data in a file format. SAP Plant Connectivity is reading the source folder where the equipment is placing the quality-data-related file. Next, SAP Plant Connectivity is sending a notification to the destination system (e.g., SAP MII) with the file content, and SAP MII is processing those data for further use.

3.1.1 Source System

In SAP Plant Connectivity, a source system specifies a system from which data needs to be queried. You can configure multiple source systems in the SAP Plant Connectivity management console. (After successful installation of SAP Plant Connectivity you can open the SAP Plant Connectivity management console which is explained in Chapter 2, Section 2.3.) You can find the SAP Plant Connectivity management console from the Windows start menu by selecting **SAP Plant Connectivity • Management Console**. From there, you'll get the source data. Agents in SAP Plant Connectivity are responsible for establishing the data flow between the data source and SAP Plant Connectivity. A source system transforms the server data in the data source into a tag-based data structure that can be browsed. It also converts the protocol-specific data types to the .NET data types used in SAP Plant Connectivity. SAP Plant Connectivity provides a list of source systems that can be used to connect with different types of data sources, as shown in Table 3.1.

Source System Type	Data Source Used
Asset Framework source system	OSIsoft PI Asset Framework, an integral part of the OSIsoft PI system.
Citect source system	Used to connect with CitectSCADA system, a control system for the monitoring and control of production plants.
File system source system	Used to monitor a specific local or remote directory of a system for data.
IP21 source system	Used to establish a connection to the Aspen InfoPlus.21 database.
Modbus source system	Used to establish a connection between SAP Plant Connectivity and PLCs or other devices.
MQTT source system	Used to establish a connection to an MQTT broker (MQTT server).
ODBC source system	Used to establish a connection to any database management system using the Open Database Connectivity (ODBC) interface.
OLEDB source system	Used to establish a connection to an OLEDB-based data source and to execute database queries.
OPC A&E source system	Used to establish a connection to OPC alarms and events server.
OPC DA source system	Used to establish a connection to OPC data access server.
OPC HDA source system	Used to establish a connection to OPC historical data access server.
OPC UA source system	Used to establish a connection to OPC unified architecture.
OSISoft PI source system	Used to establish a connection to OSIsoft PI system.
Proficy Historian source system	Used to establish a connection to GE FANUC Proficy Historian server.

Table 3.1 SAP Plant Connectivity Source Systems

Source System Type	Data Source Used
Socket source system	Used to establish a connection to a TCP/IP-based socket server.
Timer source system	A source system which changes its value periodically at specified times is treated as a timer. You can use timers to trigger actions using SAP Plant Connectivity at configurable, usually periodic, times.

Table 3.1 SAP Plant Connectivity Source Systems (Cont.)

Now you know about the available source system types in SAP Plant Connectivity and usage of those source systems. Next, let's discuss how to create a source system in SAP Plant Connectivity using an available source system type. Let's consider creation of a file system source system. In the SAP Plant Connectivity management console, you need to click the **Add Source System** button in the **Source Systems** panel. The **Add Source System** window will appear, where you need to select **File System Source System** in the **Source System Type** dropdown field. You need to enter a name and description for the source system. Then you can click the **OK** button, and the source system will be available in the **Source Systems** panel. Now click the source system from the panel and perform the required configuration in the **Settings**, **Authentication**, and **Reliable Connection** tabs, as shown in Table 3.2.

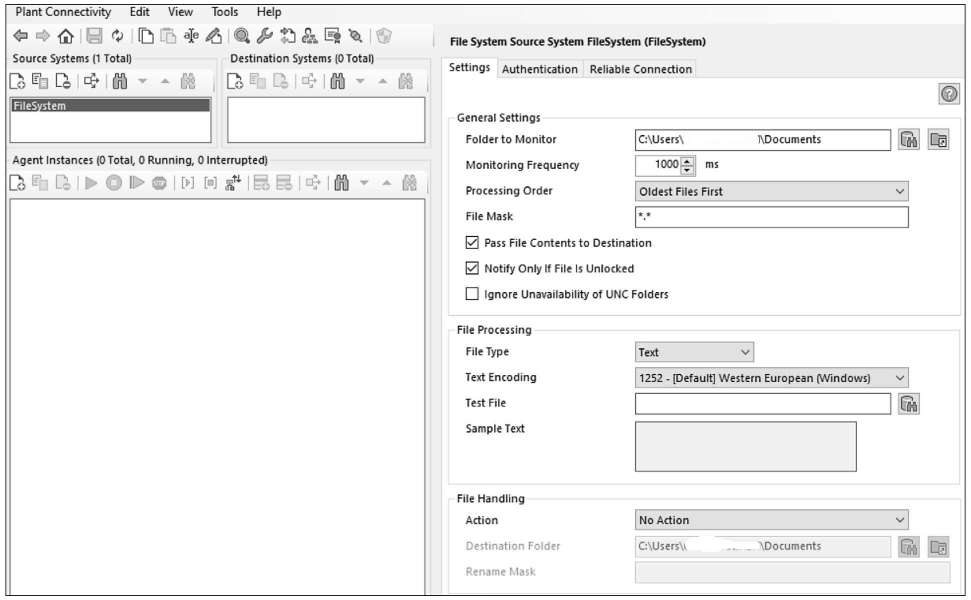
Field	Purpose
Folder to Monitor	You need to enter the folder path from where the existing file will be checked. For example, *C:\Users\Myfolder\Doc*.
Monitoring Frequency	You can specify how often (in milliseconds) the folder is to be checked for files.
Processing Order	You can select available processing options from the dropdown, such as **Random (Optimal Speed)**, **Oldest Files First**, or **Newest Files First**.
File Mask	You can enter the search criteria for the files that are relevant for processing. The default value is "*".
Pass File Contents to Destination	If you select the checkbox, then the file content will be sent to a destination system; otherwise, only the file name is sent to the destination system.
Notify Only If File Is Unlocked	If the checkbox is checked, then SAP Plant Connectivity sends a notification message if the read-only indicator for a file is not set.

Table 3.2 SAP Plant Connectivity File Source System Configuration

Field	Purpose
Ignore Unavailability of UNC Folders	If you configure a network folder with a Universal Naming Convention (UNC) path and the folder isn't accessible during runtime, then the agent gets an error status. If you check the **Ignore Unavailability of UNC Folders** checkbox, then SAP Plant Connectivity ignores this error and continues monitoring as soon as the folder is available again.
File Type	You can select the file type from the available dropdown options, such as **Text** and **Binary**.
Text Encoding	You can select text encoding from available list, then the contents of the text files are read from the source folder and the content is converted into a Unicode character string before sending to the destination systems.
Test File and Sample Text	In this field, you can use a text file to test the chosen text encoding. The file contents are displayed in the **Sample Text** field.
Action	This specifies how to handle the file after successfully reading it from the source folder. You can select a suitable option from the list, such as **No Action**, **Rename**, **Move**, **Copy**, or **Delete**.
Destination Folder	This indicates the destination folder to which the file is to be moved if the action is set to **Move** or **Rename**.
Rename Mask	In this field, you can specify the new name of the file.
Authentication Required	You can check this checkbox if authentication is required to access the folder in order to access the source file.
Domain	You can specify the Windows domain here.
User Name	You can specify a username for authentication.
Password	You can specify a password for authentication.
Max. Number of Retries	You can specify a maximum number of retries for reliable connection.
Retry Interval	You can select a retry interval in seconds for reading the file.

Table 3.2 SAP Plant Connectivity File Source System Configuration (Cont.)

After entering the required values for configuration, as shown in Figure 3.1, you can save the configuration. In this section, you learned about the process to add and configure source systems. In the next section, we'll discuss destination system configuration in SAP Plant Connectivity.



File System Source System FileSystem (FileSystem)

General Settings

Folder to Monitor: C:\Users\ \Documents

Monitoring Frequency: 1000 ms

Processing Order: Oldest Files First

File Mask: *.*

☑ Pass File Contents to Destination

☑ Notify Only If File Is Unlocked

□ Ignore Unavailability of UNC Folders

File Processing

File Type: Text

Text Encoding: 1252 - [Default] Western European (Windows)

Test File:

Sample Text:

File Handling

Action: No Action

Destination Folder: C:\Users\ \Documents

Rename Mask:

Figure 3.1 Configuring File System Source System

3.1.2 Destination System

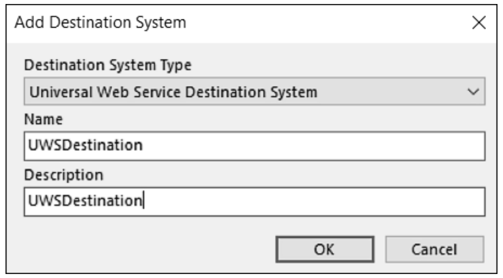
In SAP Plant Connectivity, the destination system is the system to which the notification is sent based on the condition and notification process. In the **Destination Systems** panel, you can add multiple destination systems and send notifications to appropriate destination systems. SAP Plant Connectivity provides a list of different types of destination systems, as shown in Table 3.3.

Destination System Type	Purpose of Destination System Type
MQTT Destination System	The MQTT destination system is used to send MQTT messages to an MQTT broker. It allows publishing data that has been transferred to SAP Plant Connectivity to an MQTT broker.
Multiple Call Destination System	This is used to call another destination system in a configurable sequence. A multiple call destination (MCD) system also allows other functions that can be combined with each other.
ODBC Destination System	Using an ODBC destination system, you'll be able to connect an ODBC database to SAP Plant Connectivity and you can send notification messages to this database. SAP Plant Connectivity supports connecting to the SAP HANA database and MS SQL Server ODBC database.
OPC UA Destination System	The OPC UA destination system allows you to connect an OPC UA server. You can use OPC UA destination systems for tag-based notifications and method notifications.
Query Destination System	This allows you to retrieve current tag values from the source system of an agent instance or to store tag values to the source system of an agent instance. The agent instance can be the same agent instance from which the query destination system is called, or it can be another agent instance available in SAP Plant Connectivity.
RFC Destination System	This allows you to connect SAP NetWeaver, OPC data access, and SAP EWM.
Simulation Destination System	The simulation destination system is used for testing the notification process very quickly. It's used for testing by sending notification messages to a specific folder.
Universal Web Service Destination System	The universal web service destination system allows you to send notification messages to web services (WSDL), OData services, and RESTful web services. You can also send notifications from SAP Plant Connectivity to SAP MII using a universal web service destination system.

Table 3.3 Destination System Type

Table 3.3 Destination System Type (Cont.)

To add a destination system in the **Destination Systems** panel, you need to click the **Add Destination System** button in the **Destination Systems** panel. The system will show a new window to add a destination system where you need to select the appropriate type of destination system from the **Destination Systems Type** dropdown (e.g., **Universal Web Service Destination System**). Now you can enter a name and description for the destination system in **Name** and **Description** fields. Finally, you can click the **OK** button, as shown in Figure 3.2.



Add Destination System

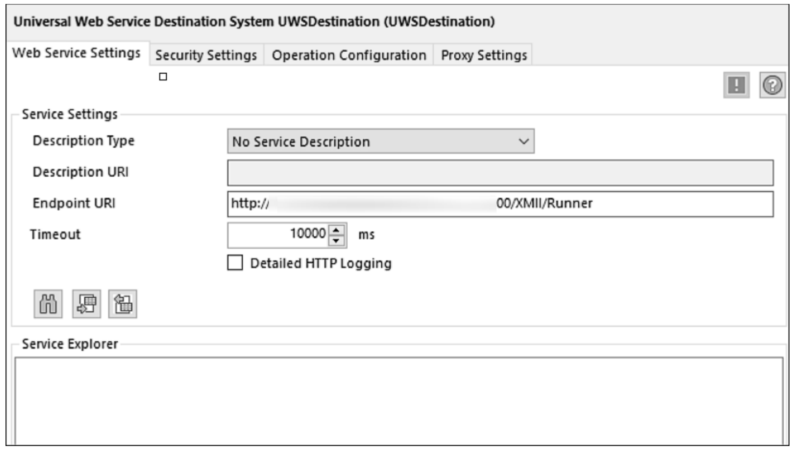
Destination System Type: Universal Web Service Destination System

Name: UWSDestination

Description: UWSDestination

Figure 3.2 Adding SAP Plant Connectivity Destination System

After adding the destination system, you can perform the required configuration for the destination system. We'll use the example of configuring a universal web service destination system to send the file system source system file data to the target SAP MII system. In the **Web Service Settings** tab, choose the description type from the available options, such as **No Service Description** (used for RESTful Web services), **WSDL**, and **OData**. To configure the RESTful web services setting for the SAP MII system, select **No Service Description** value for the **Description Type** field and provide the "https://<server name>:<port number>/XMII/Runner" URI in the **Endpoint URI** field, as shown in Figure 3.3.


Universal Web Service Destination System UWSDestination (UWSDestination)

Service Settings

Description Type: No Service Description

Description URI:

Endpoint URI: http://[redacted]00/XMII/Runner

Timeout: 10000 ms

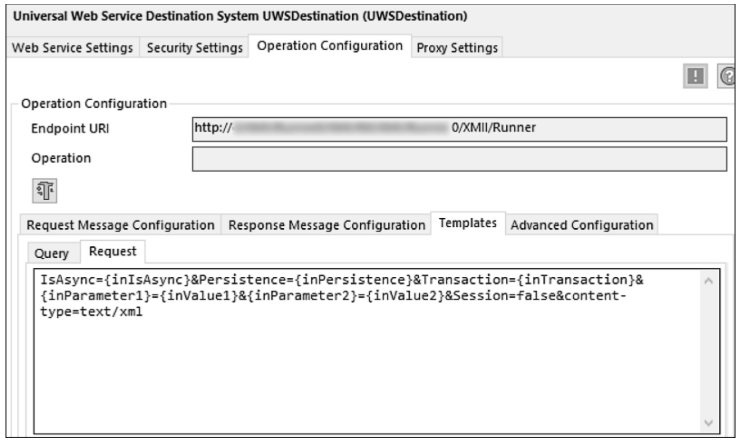
☐ Detailed HTTP Logging

Service Explorer

Figure 3.3 Configuring Endpoint URI for SAP MII

On the **Security Settings** tab, you can select the appropriate authentication method and perform the required parameter configurations. On the **Operation Configuration** tab, you can select **POST** as the method. Then move to the **Templates • Request** tab, where you can enter the template, as shown in Figure 3.4. On the **Request Message Configuration** tab, you can find the input variables that appear from values in the curly brackets of the template request. The system interprets the values inside the curly brackets as input variables, which you can find in the **Request Message Configuration** tab. You can assign a **True** or **False** value to the **inIsAsync** variable, which indicates whether the SAP MII transaction can be processed asynchronously. You can set a value of **NEVER**, **ALWAYS**, or **ONERROR** for the **inPersistence** variable, which is used to specify the persistency control of the SAP MII transaction.

In the **inTransaction** variable, you can specify the name of SAP MII transaction—for example, Default/Chandan/Utility/BLS_PCoIntegration. You can specify the first SAP MII transaction parameter name in the **inParameter1** variable and you can also specify the value of the first parameter in the **inValue1** variable.


Universal Web Service Destination System UWSDestination (UWSDestination)

Operation Configuration

Endpoint URI: http://[redacted]0/XMII/Runner

Operation:

```
IsAsync={inIsAsync}&Persistence={inPersistence}&Transaction={inTransaction}&
{inParameter1}={inValue1}&{inParameter2}={inValue2}&Session=false&content-
type=text/xml
```

Figure 3.4 Operation Configuration for Universal Web Service Destination System

3.1.3 Agent Instance

In SAP Plant Connectivity, the agent instance plays a key component role that's used to establish the connection between the source system and SAP Plant Connectivity. It also helps to enable processing and forwarding messages to the destination system. An agent instance is created based on one source system; it's also possible to create an agent instance without a source system, in which case the agent instance can be used as an OPC UA (unified architecture) server or as a web server. Each agent instance is executed as a standalone windows service, and you can run multiple agent instances at the same time, each of which is executed independently. Now let's see how to create an agent instance in SAP Plant Connectivity.

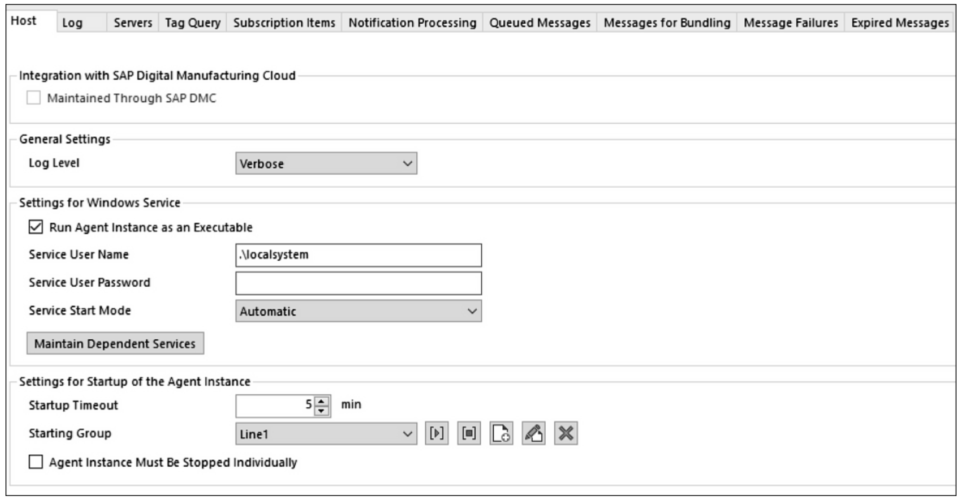
To create an agent instance, first click the **Add Agent Instance** button under the **Agent Instances** panel. The system will display a popup window where you need to select the source system from available dropdown list. Enter the instance name and instance description, then click the **OK** button. The agent instance will be available under the **Agent Instances** panel. Next you need to perform the required configurations for the created agent instance, as we'll discuss in the following sections.

Host

Before starting the agent instance, you need to establish the settings for the host in the **Host** tab of the agent instance. You won't be able to change the configuration in the **Host** tab if the agent instance is in a running state. In the **Host** tab, select a **Log Level** from the available values, such as **Off**, **Critical**, **Error**, **Warning**, **Information**, and **Verbose**. Selecting the **Run Agent Instance as an Executable** checkbox allows the agent instance to run as a normal windows process with the rights of the logged-in user account. It runs as the user logged on in Windows and it stops if the user logs off. If the

Run Agent Instance as an Executable checkbox isn't checked, then the agent instance runs as a Windows service.

If the agent instance needs to be run as a Windows service, then you need to specify a service username, password, and start mode in the **Service User Name**, **Service User Password**, and **Service Start Mode** fields. The **Maintained Dependent Services** button is used to maintain any dependent service that can be started before the agent instance is started. The **Startup Timeout** field allows you to set the maximum time in minutes taken to start the agent instance. If the agent instance isn't started by that time, then the start process is terminated. In the **Starting Group** field, you can assign an agent instance to an existing starting group or you can create a starting group using the **Create New Starting Group** button and assign it to an agent instance, as shown in Figure 3.5. Using a starting group, you can start and stop multiple agent instances together that belong to the same starting group. If you want to stop an agent instance individually, then you can select the **Agent Instance Must Be Stopped Individually** checkbox.



Integration with SAP Digital Manufacturing Cloud

□ Maintained Through SAP DMC

General Settings

Log Level: Verbose

Settings for Windows Service

☑ Run Agent Instance as an Executable

Service User Name: .\localsystem

Service User Password:

Service Start Mode: Automatic

Settings for Startup of the Agent Instance

Startup Timeout: 5 min

Starting Group: Line1

□ Agent Instance Must Be Stopped Individually

Figure 3.5 Configuring SAP Plant Connectivity Agent Instance Host

Log

In the **Log** tab, you can search the log entries for the agent instance. SAP Plant Connectivity writes separate logs for individual agent instances, and log data are saved in the Windows event log. You can configure the required fields in the **Log** tab to restrict the search for log data, as shown in Table 3.4.

Field Name	Purpose
Start Date	Start date and time for searching log entries.
End Date	End date and time for searching log entries.
Filter Level	Refers to the log type that will be displayed in a search. Available values for filter level are **Critical**, **Error**, **Warning**, **Information**, and **Verbose**.
Number of Entries	Lets you set the number of log entries to display. Available values are **1000**, **2000**, **5000**, and **All Entries**.

Table 3.4 Agent Instance Log Configuration

Table 3.4 Agent Instance Log Configuration (Cont.)

After selecting the date and filter conditions, click the **Refresh** button. The log entries are displayed in a table, in descending order based on the time of the logged events. You can delete all log entries by clicking the **Delete All Log Entries** button. Using the **Export to CSV File** button, you can export searched log entries in CSV file into a folder as per your selection. By clicking the **Manage Windows Event Log** button, you can start the Windows Event Viewer for the Windows log assigned to the agent instance.

At the bottom of the log entries table, you can use the **Search For** field to select displayed log entries that match specific search criteria. The available values in the **Search For** field are **Messages**, **Event ID**, **Event Type**, and **Source**. You can select one of the values from the field and click the **Execute Search** button to select appropriate log entries based on the search criteria. **First Selected Log Entry**, **Previous Selected Log Entry**, **Next Selected Log Entry**, **Last Selected Log Entry**, **Select Log Entries with Same Thread IDs**, and **Copy Selected Log Entries to Clipboard** buttons are also available to perform manual functions on selected individual or multiple log entries.

Servers

In the **Servers** tab of agent instance, you can configure various server types, which are used for different query interfaces. The available server types you can select for agent instances are listed in Table 3.5.

Server Type	Required Settings
SAP MII Query Server	Used where you're using SAP MII 12.2 or higher and you want to connect with SAP Plant Connectivity for processing the tag query or PCo query. To perform the server configuration for an agent instance, you need to check the **SAP MII Query Server** checkbox in the **Available Server Type** list in the **Servers** tab. You can specify the port for connection in SAP MII in the **Port** field. In the **Security** field, you can choose an available value, such as **Unsecured Connection** or **Connection with TLS**. In the **Server Certificate** field, you can specify the certificate used by SAP Plant Connectivity to establish a secure connection with SAP MII.

Table 3.5 SAP Plant Connectivity Agent Instance Server Type List

Server Type	Required Settings
SAP MII Query Server (Before 12.2)	Used where you're using an SAP MII system version lower than 12.2 and you want to connect with SAP Plant Connectivity for processing a tag query or PCo query. To perform the server configuration for an agent instance, you need to check the **SAP MII Query Server** checkbox in the **Available Server Type** list in the **Servers** tab. You can specify the port for connection in SAP MII in **Port** field (the same port has to be specified for the PCoConnector data server configured in SAP MII). One port is to be used for one agent instance only. In the **Include Quality** field, you can select one value from the available options, such as **No Quality Information**, **Quality Information as String**, **Numeric Quality Information**, and **Quality Information as String and Numeric**. The value in the **Include Quality** field indicates how quality information is to be provided for the measurement values that are to be transmitted. There are two checkboxes for UDS compatibility: **Calculate Minimum and Maximum for Intervals without Value Changes** and **Assign Statistical Values to End of Interval**. The latter indicates that the calculated statistical values are always assigned to the end of the evaluation interval.
SAP NW RFC Server	If you need to connect SAP Plant Connectivity with SAP S/4HANA for query processing, then you can set up an SAP NetWeaver RFC server in the **Servers** tab of the selected agent instance. To establish the connection, you need to fill in the following fields: **Registration Using Message Server**, **SNC**, or **Tracing and Repository**. You need to provide values to establish the connection, in the fields **Program ID**, **SAP Gateway Host**, **SAP Gateway Service**, **SAP Router**, and so on. You can test the connection by clicking the **Test Registration** button.
SAP ODA RFC Server	If you need to achieve Open Platform Communications (OPC) Data Access functionality in an SAP S/4HANA system then you can select **OPC Data Access RFC Server** in the **Servers** tab of the agent instance. The setting is only possible in connection with an OPC Data Access agent. Using the **Test Registration** button, you can test the connection. You need to enter the values for **Program ID**, **SAP Gateway Host**, and **SAP Gateway Service**; when using load balancing, you need to enter values for **SAP Message Server Host**, **SAP System ID**, and **SAP Logon Group**.
SAP EWM RFC Server	Required when you want to connect an SAP EWM system with SAP Plant Connectivity to send telegrams. You need to provide required information to establish connection with the SAP EWM system. Using the **Test Registration** button, you can test the registration; you need to configure required fields like **Program ID**, **SAP Gateway Host**, **SAP Gateway Service**, or **SAP Message Server Host**, **SAP Logon Group**, and **SAP System ID** if you're using load balancing.

Table 3.5 SAP Plant Connectivity Agent Instance Server Type List (Cont.)

Server Type	Required Settings
WebSocket Server	Used when you want SAP Plant Connectivity to act as a WebSocket server for web applications. The notification process and query process are supported by the WebSocket server. In the notification process, the events provided by the source system can be displayed in the UI. In the query process, the client can send messages to the SAP Plant Connectivity WebSocket server. You need to enter required details for configuring WebSocket settings, such as **Protocol**, **Port**, **Server Certificate**, **Message Format**, and so on. You can also maintain the destination system by clicking the **Created Destination System** button. Using the **Delete Destination System** button, you can delete a destination also. The status of the created destination system will be displayed in the **Status** field.
OPC UA Server	Used if you want to run an agent instance as an OPC UA server. In the **OPC UA Server Settings** tab, you can define the server endpoint. In the **Security Configuration** tab, you can configure secure connections and user authentication for OPC UA servers. In the **Server Method Definitions** tab, you can add new method definitions to the server.
PCo Web Server	If you want to run an SAP Plant Connectivity agent instance as a web server, then you can use this checkbox to configure the web server. In the **PCo Web Server Settings** tab, you can add an end server endpoint, and you can define a web server method in the **Web Server Method Definitions** tab.

Table 3.5 SAP Plant Connectivity Agent Instance Server Type List (Cont.)

Tag Query

In the query process, required tags can be specified in the **Tag Query** tab. You can establish the cache settings in the **Tag Query** tab with the **Cache Mode**, **Mask**, and **Alias** fields. The **Cache Mode** field provides four options you can choose from per your requirements:

- **Access to Data Source Only**
  Allows you to retrieve directly from the data source for queries.
- **Access to Cache, to Data Source as Required**
  Allows you to check the cache first to see whether there are tags and metadata. If not, then the tags are retrieved from the data source and added to the cache.
- **Access Using Aliases Only**
  Used if you previously created an alias for the source system.
- **Access to Cache Only**
  Used to query all tags from the data source that are stored in the cache.

Subscription Items

In the **Subscription Items** tab, you can subscribe to tags from the source system; these subscriptions can be used for notification processes by the agent instance. You can also change subscriptions or delete subscription items from the list. You can browse multiple tags from the source system by clicking the **Browse for Tags** button in the **Subscription Items** tab. You can also add individual tags by clicking the **Add New Subscription Item Manually** button. The subscription items will be displayed in a table inside the **Subscription Items** tab, and the table contains fields like **Name**, **Source**, **Deadband**, **Only Changes**, and **Data Type**, as described in Table 3.6.

Field Name	Purpose
Name	This is the name of the tag.
Source	This is the tag name in the namespace of source system.
Deadband	In the SAP Plant Connectivity notification process, deadband refers the range of allowed variance in values and is used to filter out disallowed measurement values. For example, let's say you have specified a deadband value of "2" in the **Subscription Items** tab. The last tag value was 52 and next tag value is 54.5. The difference between these two values is 2.5 (53.5 – 52); this is higher than the specified deadband value 2. As such, SAP Plant Connectivity will send a notification to the destination system. If the difference between the current tag value and the next tag value had been less than specified deadband value then SAP Plant Connectivity would not have sent a notification to destination system.
Only Changes	For a tag, if you select the **Only Changes** checkbox then SAP Plant Connectivity sends a notification to the destination system if the tag value is changed compared to last value of the same tag; otherwise, no notification will be sent.
Data Type	Refers to the data type of the tag. The data type of the tag is determined automatically during browsing for tags. If the data type isn't supported by SAP Plant Connectivity, then the tag can't be selected during browsing.

Table 3.6 SAP Plant Connectivity Agent Instance Subscription Items

You can also change the attribute of a subscribed tag by selecting the tag's row from the subscription item list and clicking the **Edit Subscription Item** button.

Notification Processing

In the **Notification Processing** tab, you can perform two types of activities for notification messages: **Notification Message Queue and Dispatch Settings** and **Enhanced Notification Processing**. In the notification process, before sending messages to the destination system, the messages are placed in a queue, which helps avoid blocking the processing of further notifications from source system. It also helps to start a delivery retry if delivery fails. Individual message queues are created automatically for each agent instance, and queues are deleted when the agent instance is deleted. Different types of queues are created for each agent instance, such as the main queue, queue for message bundling, failed message queue, and expired message queue. In the **Notification Message Queue and Dispatch Settings** section, you can establish the required settings, as explained in Table 3.7.

Field Name	Purpose
Storage Method	Refers to how notification messages can be stored in the queue. The available options are **In Memory Only**, **File System**, and **Microsoft Message Queuing (MSMQ)**.
Keep Messages in Memory	This checkbox selection option is available for the file system storage method only. It's autoselected for the other two storage methods. If this checkbox is selected, then all messages are stored in SAP Plant Connectivity memory.
Resend Failed Messages Automatically	If the checkbox is selected, then the system tries to resend the failed messages automatically. If the checkbox isn't selected, then you must perform manual resending of messages from the **Message Failure Tab**.
Keep Expired Messages	Selection of this checkbox specifies that expired notification messages are moved to the expired messages queue. You can find the expired messages on the **Expired Messages** tab. If the messages aren't deleted from the expired messages queue, then that queue grows continuously. If the checkbox isn't selected, then notification messages are deleted automatically from all queues after expiry.
Process Notification Messages Exactly Once in Order	Indicates all notification messages are delivered individually and exactly once in the order in which they were placed in the queue.
Keep Copies of Queued Notification Messages in Journal Queue	Selection of this checkbox specifies that a copy of a notification message is stored in the **Journal Messages** folder of MSMQ. The system will allow you to select the checkbox if you selected the MSMQ storage message. Keep in mind that this option should be used for testing purposes only; otherwise, it can affect system performance.
Make Queued Notification Messages Recoverable	The system allows you to select this checkbox if you selected the MSMQ storage method. If checked, notification messages are stored on the system hard disk and are available after a Windows restart. Notification messages are saved in the working memory if the checkbox isn't selected.

Table 3.7 SAP Plant Connectivity Agent Instance Notification Processing

Field Name	Purpose
Max. Queued Messages	This field allows you to set the value for the maximum number of messages to be in the message queue; this includes the messages in the **Queued Messages** tab and in the **Message Failures** tab.
Max. Dispatch Threads	You can enter the value for the maximum number of threads per CPU core for parallel sending of notifications. The minimum and maximum range is 1 to 250 per CPU core. The value you enter will be multiplied by 8 (because your SAP Plant Connectivity system has eight CPU cores). You need to keep in mind that this setting isn't applicable for the **Process Notification Messages Exactly Once in Order** option. If the **Process Notification Messages Exactly Once in Order** checkbox is checked, then the **Max. Dispatch Threads** field will be disabled.

Table 3.7 SAP Plant Connectivity Agent Instance Notification Processing (Cont.)

In the **Enhanced Notification Processing** section, you can configure the enhancement of the notification process. Based on the selection of available radio buttons, different options will be available for enhancement as described in Table 3.8.

Field Name	Purpose
No Enhanced Notification Processing	You can select the radio button if there is no requirement for enhanced notification.
Destination System Calls with Response Processing	This is the SAP-standard enhanced notification process available with an SAP Plant Connectivity installation. If you select the radio button, then the SAP Plant Connectivity standard enhanced notification is tagged automatically. After selecting the radio button, you need to create the destination system by clicking the **Create Destination System** button in the **Maintain Destination System for Notification Enhancement** section.
Automatic Configuration Backup	You can select this option if you want to create a data backup of the configuration. To setup automatic data backup, select the menu path **Plant Connectivity • Set Up Automatic Backup**. The system will show a popup window where it creates a source system and an agent instance with the name AutoBackupConfig. Then you can set the execution plan for the timer in the _AutoBackupConfig source system and start the agent instance. In the **Notification Processing** tab of any agent instance, select the **Automatic Configuration Backup** radio button. Then, you can create the destination system for this notification processing by clicking the **Create Destination System** button.

Table 3.8 SAP Plant Connectivity Agent Instance Enhanced Notification Processing

Field Name	Purpose
Automatic Configuration Backup (Cont.)	Next, you need to create a static notification for the agent instance, and in **Output** tab of the notification, generate the output expression by clicking the **Generate Expressions** button. In the **Destinations** tab, you can assign the **/Enhanced Notification Processing** destination system to the notification.
Customer-Owned Enhancement	Using this option, you can create your own enhancement where you need to create your own dynamic link library (DLL) and enter it in the **Dynamic Link Library** field by clicking the **Browse** button. If you have only one class file that implements the enhanced notification processing interface, then that class appears automatically in the **Class** field by selection of the DLL file.

Table 3.8 SAP Plant Connectivity Agent Instance Enhanced Notification Processing (Cont.)

Note

Enhanced notification processing allows you to control and document the data flow flexibly in production in connection with one or more destination systems one after the other in any order—for example, a web service, a RESTful web service, an OData destination, or an ODBC destination. SAP also delivers the standard enhancement destination system calls with response processing for enhanced notification processing. SAP Plant Connectivity allows you to execute one or multiple destination system calls and you can write back the results of the calls to the data source of the agent instance or another agent instance.

Queued Messages

In the **Queued Messages** tab, you can find all notification messages currently queued. You need to click the **Refresh** button to get the queued message list in this tab.

Messages for Bundling

In **Messages for Bundling** tab, you'll be able to find the messages that are to be bundled by clicking the **Refresh** button. To get the messages in the **Messages for Bundling** tab, you need to configure message bundling on the **Message Delivery** tab of the notifications in the agent instance. In the message bundling concept, you can specify that instead of sending individual messages to a destination system, you'll either send a fixed number of messages in a bundle, or set a maximum time in seconds to wait to accumulate the messages that will be bundled into a single message to be sent to the destination system.

Example

Consider a scenario where a high-speed shop floor machine is generating data in second and that same data needs to be sent to SAP MII through notifications. You can bundle the messages per your requirements and configure a fixed number of messages or maximum accumulation time, then we can send the bundled messages to the SAP MII system in a single call instead of making multiple calls to the SAP MII destination system. This can increase system performance.

Message Failures

In the **Message Failures** tab, you need to click the **Refresh** button to see the failed messages that haven't been sent successfully even after multiple attempts. You can send one or more messages by clicking the **Resend All** button and you can delete messages by clicking the **Delete All Messages** button.

Expired Messages

In the **Expired Messages** tab, you can see the messages that couldn't be sent and that are past the predefined lifetime. You need to click the **Refresh** button to see the message list, and you can delete messages by clicking **Delete All Messages**.

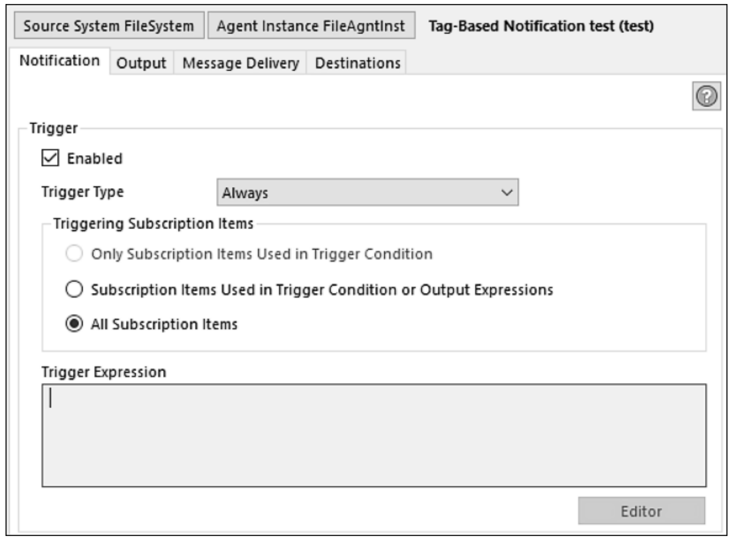
3.1.4 Notification

A notification is defined under the agent instance to send message information to the destination system based on the occurrence of a specific event. To add a notification under the agent instance, you need to select the agent instance and click the **Add Notification** button. The system will show the options for notification types such as **Static Notification**, **Notification Template for Remote Subscriptions**, **Versioned Notification**, and **Method Notification**. Static notifications are in the tag-based notification category. Static notifications are treated as normal notifications. The notification template is a copy template for notification messages; you need to create the template for remote subscription. The versioned notification contains one or multiple notifications with different versions, and notification versions can be draft, active, or expiring versions. You can select a method notification option when you want to activate a method of the OPC UA server or SAP Plant Connectivity web server, and you can define which action you want to run if the method is called by a client.

Example

Consider the scenario in which an agent instance is created based on a file system source system in SAP Plant Connectivity. Shop floor equipment is generating the quality-related data for a specific product and placing those data in a file. Now the created notification under same agent instance will send the file content to the destination system (e.g., SAP MII) based on the trigger type and condition which you can find in later in this section. You can send the notification in XML format or you can send the values for specific fields that will be processed in SAP MII.

Once the notification is added under the agent instance, you need to configure the notification, which will send an XML message, defined in the output configuration, to the destination system (e.g., SAP MII). You need to create a notification for the agent instance as shown in Figure 3.6. In the **Notification** tab, you can specify the *trigger expression* for the notification—that is, the condition based on which the notification message will be generated. You can specify **Always** in the **Trigger Type** dropdown, which will generate the notification always whenever there is a change of the value in the source system. In that case, you don't need to specify a trigger expression. You can also select **OnTrue**, **OnFalse**, **WhileTrue**, or **WhileFalse** under **Trigger Type**. **WhileTrue** or **WhileFalse** will send a notification message every time the value changes and the expression evaluates to true or false, respectively, but for **OnTrue** and **OnFalse**, it will send the notification message once when the expression to be evaluated evaluates to true or false, respectively, and won't send it again unless the expression evaluation result changes.



Trigger

☑ Enabled

Trigger Type: Always

Triggering Subscription Items

○ Only Subscription Items Used in Trigger Condition

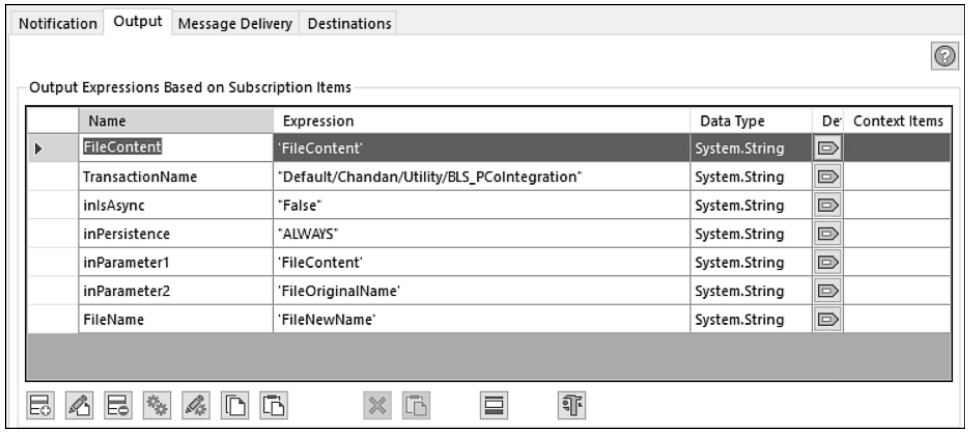
○ Subscription Items Used in Trigger Condition or Output Expressions

● All Subscription Items

Trigger Expression

Figure 3.6 Configuring SAP Plant Connectivity Agent Instance Notification

In the **Output** tab, you can define the information that is to be passed to a destination system. The output values are added based on subscription items, as shown in Figure 3.7. You can add any number of elements in the output message to specify the actual subscription item values from the source system, a timestamp, or any other fixed values. The subscription items can be added in the output expression within single quotes. Any fixed value can be added within double quotes. Figure 3.7 shows the output configuration for sending file content to the SAP MII system.



Output Expressions Based on Subscription Items

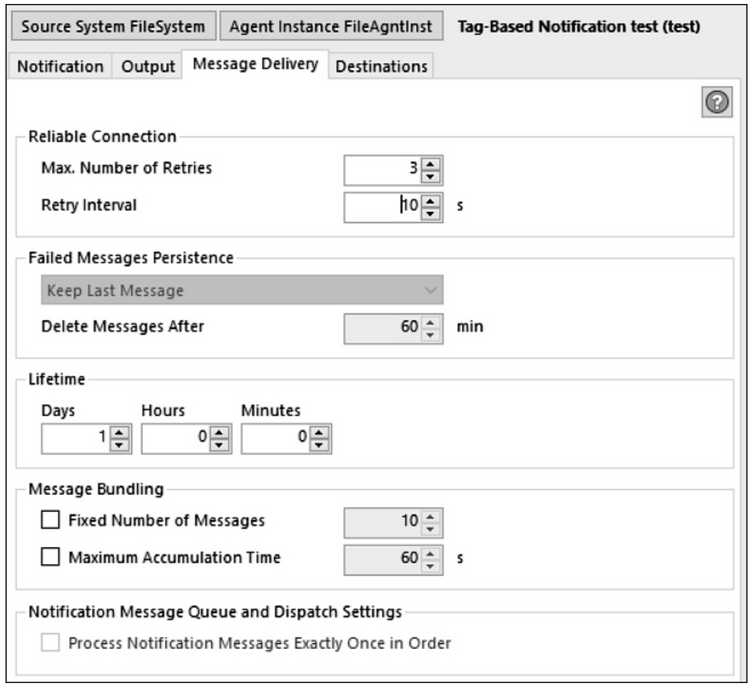
	Name	Expression	Data Type	De	Context Items
▶	FileContent	'FileContent'	System.String	⊏	
	TransactionName	"Default/Chandan/Utility/BLS_PCoIntegration"	System.String	⊏	
	inIsAsync	"False"	System.String	⊏	
	inPersistence	"ALWAYS"	System.String	⊏	
	inParameter1	'FileContent'	System.String	⊏	
	inParameter2	'FileOriginalName'	System.String	⊏	
	FileName	'FileNewName'	System.String	⊏	

Figure 3.7 Configuring Output for SAP Plant Connectivity Agent Instance Notification

In the **Message Delivery** tab, you can configure how to deliver messages to the destination system. In the **Reliable Connection** section, you can configure values for **Max. Number of Retries** and **Retry Interval (Seconds)**. In the **Failed Message Persistence** section, you can select values for how failed notification messages are to be stored, and you can enter the value in minute for the **Delete Messages After** field. All failed messages older than the value in **Delete Messages After** are to be deleted. In the **Lifetime** section, you can specify how long the messages are to be kept in the queue. For many of these scenarios, the frequency of the data coming from the machine or control system may be very high (e.g., a few hundred messages in a minute), so sending each of them individually to a destination system such as SAP MII may be a huge source of performance overhead. You can configure to bundle the notification messages in the **Message Delivery** configuration tab of the notification, as shown in Figure 3.8. You can specify either a fixed number of messages to bundle or a maximum time in seconds to wait to accumulate the messages, which will be bundled into a single message to be sent to the destination system. Note that message bundling is not supported for enhanced notification processing or query destination.

Finally, you need to perform configuration for the **Destinations** tab to specify the destination where the notification message will be sent. SAP Plant Connectivity can send data to different types of destination systems, as you know from the discussion of the creation of different destination systems in Section 3.1.2. The **Destinations** tab is used to configure a notification and to configure a notification template. To add a destination, you can click the **Add Destination System** icon. The **Add Destination System to Notification** popup will appear, where you can select the destination system that you want to use for notification and enter a **Name** and **Description**. You can add a destination for a Universal Web Service destination system that will be used to send notifications to the

SAP MII system, as shown in Figure 3.9. You can then perform assignment of the destination system input variables to output values.



Reliable Connection

Max. Number of Retries 3

Retry Interval 10 s

Failed Messages Persistence

Keep Last Message

Delete Messages After 60 min

Lifetime

Days	Hours	Minutes
1	0	0

Message Bundling

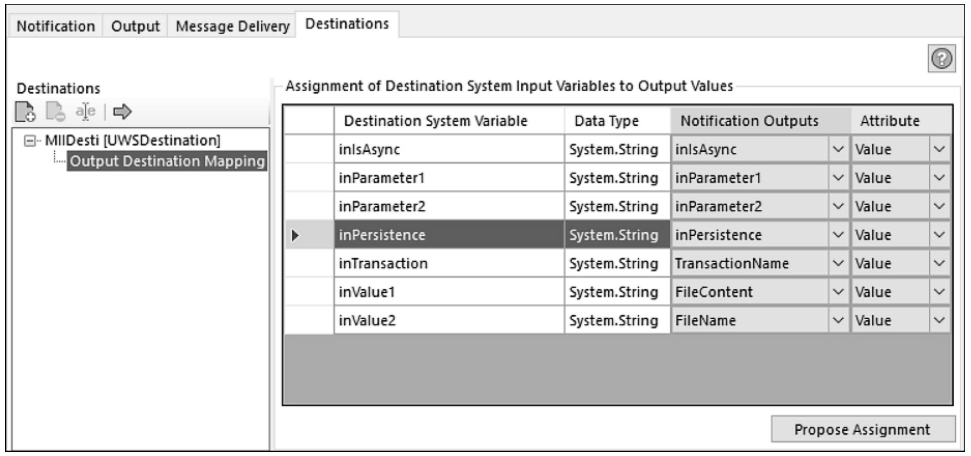
□ Fixed Number of Messages 10

□ Maximum Accumulation Time 60 s

Notification Message Queue and Dispatch Settings

□ Process Notification Messages Exactly Once in Order

Figure 3.8 Configuring Message Delivery



Assignment of Destination System Input Variables to Output Values

	Destination System Variable	Data Type	Notification Outputs	Attribute
	inIsAsync	System.String	inIsAsync	Value
	inParameter1	System.String	inParameter1	Value
	inParameter2	System.String	inParameter2	Value
▶	inPersistence	System.String	inPersistence	Value
	inTransaction	System.String	TransactionName	Value
	inValue1	System.String	FileContent	Value
	inValue2	System.String	FileName	Value

Figure 3.9 Configuring SAP Plant Connectivity Notification Destination

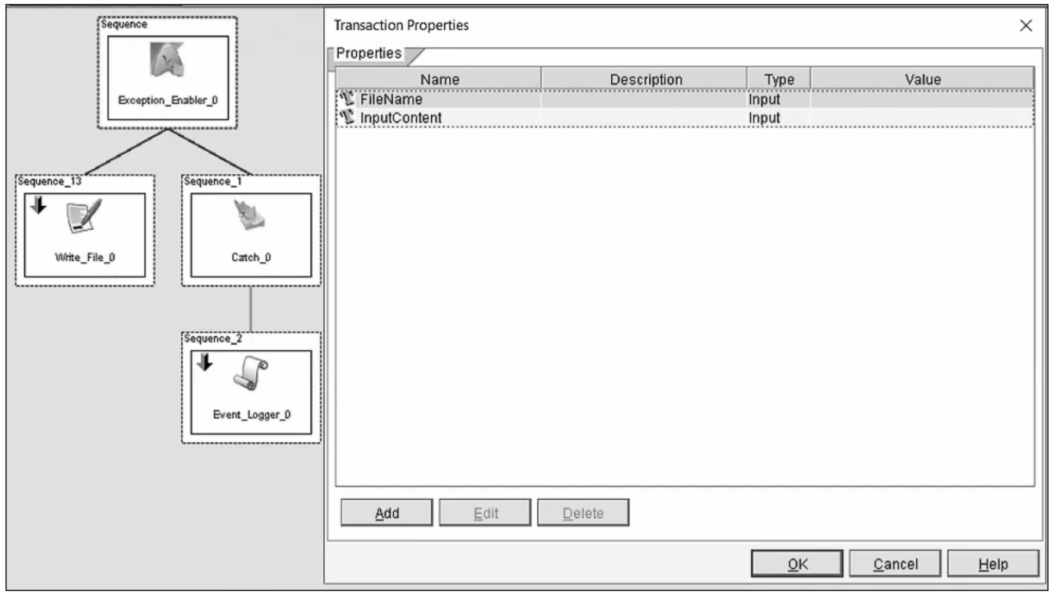
You can also send the notifications to the destination system (e.g., SAP MII) as XML, and SAP MII can process the XML notifications. Hence, you need to define an input parameter (e.g., **InputXML**) of type XML in SAP MII in **Transaction Properties**. Now, the same input parameter needs to be mapped to the **inParameter1** variable (e.g., in the **Expression** field you can map **InputXML**) in the **Output** tab of the agent notification. Next, in the **Destination** tab, you can assign **~NotificationOutputInXMLFormat** to the **inValue1** destination system variable. You can find a sample XML notification received in SAP MII from SAP Plant Connectivity in Listing 3.1.

```
<?xml version="1.0" encoding="UTF-8"?>
<NotificationMessage>
    <Header>
       <Name>Test_Notification</Name>
       <Description><![CDATA[Test_Notification]]></Description>
       <Destination>MII_DEST</Destination>
       <CreatedDate>2022-10-19T12:46:44</CreatedDate>
       <ID>2cd33216-1b4e-45bd-82f6-caf5fe673818</ID>
       <Status>Created</Status>
    </Header>
    <Body>
       <Values>
          <FileContent type="xsd:string" quality="None" timeStamp="0001-01-
01T00:00:00.0000000"><![CDATA[AHJAAHJS]]></FileContent>
          <TrasactionName type="xsd:string" quality="None" timeStamp="0001-01-
01T00:00:00.0000000"><![CDATA[Default/Chandan/BLS/BLS_TestPcoFile]]>
          </TrasactionName>
          <inIsAsync type="xsd:string" quality="None" timeStamp="0001-01-
01T00:00:00.0000000"><![CDATA[False]]></inIsAsync>
          <inPersistence type="xsd:string" quality="None" timeStamp="0001-01-
01T00:00:00.0000000"><![CDATA[ALWAYS]]></inPersistence>
          <inParameter1 type="xsd:string" quality="None" timeStamp="0001-01-
01T00:00:00.0000000"><![CDATA[InputContent]]></inParameter1>
          <inParameter2 type="xsd:string" quality="None" timeStamp="0001-01-
01T00:00:00.0000000"><![CDATA[FileName]]></inParameter2>
          <FileName type="xsd:string" quality="None" timeStamp="0001-01-
01T00:00:00.0000000"><![CDATA[dkjdkjdkjdkj]]></FileName>
       </Values>
    </Body>
    <Faults />
</NotificationMessage>
```

Listing 3.1 Sample SAP Plant Connectivity XML Notification Received in SAP MII

Once the configuration is complete, you need to start the agent instance from the SAP Plant Connectivity console, which will start the service to listen to the file system source. If you place a file in a monitoring folder, then the system will read the file, and then a notification message will be sent to the destination system for further processing. Starting the agent instance is also required to query the data source from SAP MII.

A sample notification message sent to SAP MII is shown in Figure 3.10. You can develop custom logic using the BLS transactions in SAP MII to process the notification message and initiate further processing to store the data in database, trigger alerts or user actions, or update enterprise systems.



Name	Description	Type	Value
FileName		Input	
InputContent		Input	

Figure 3.10 SAP MII Transaction for Notification Message from SAP Plant Connectivity

3.1.5 Managing Notifications from SAP MII

In the previous section, you learned about the required configurations in SAP Plant Connectivity for notification. Now you'll learn how notifications are managed in SAP MII. You can directly create and configure notifications in SAP MII and publish the notifications to SAP Plant Connectivity. SAP Plant Connectivity connects to a data source and can handle the notifications published. You can handle notifications for multiple SAP Plant Connectivity installations from a central SAP MII system. Notifications can be paused or modified from SAP MII. SAP MII provides a UI for notification management, which is available in SAP MII by following menu path **Plant Connectivity Integration • Notification Management**. When you click the **Notification Management** link, the system will open a UI with the URL *http://<Host>:<Port>/XMII/PcoManagement.jsp* that contains the **Manage Notifications by SAP Plant Connectivity**, **Manage Notifications by PIC**, and **Browse Notifications** activities, as shown Figure 3.11.

Using notification management functionality in SAP MII, you can create and configure notifications that are processed on the SAP Plant Connectivity side. Based on actions such as create, modify, and publish, the status of the notification is changed in SAP MII and in SAP Plant Connectivity. In SAP MII, when you change an existing notification with the status set to **Current** or **Current-Paused**, then a new version is created. If you edit a notification with the status set to **Current** from SAP MII, then a new version of the notification with the status set to **Draft** is created in SAP MII and the notification status of **Current** is changed to **Outdated**. In SAP Plant Connectivity, a new version of the notification with the status **Draft** is created and the status of an existing notification is changed to **Expiring**. If you edit a notification with the status **Current-Paused**, then a new version of the notification will be created with status **Draft**. After publishing and activating a notification, the status is changed to **Current-Paused**.

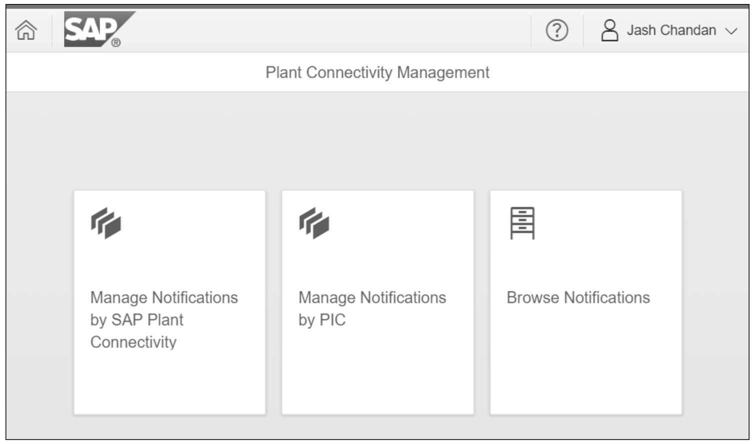

SAP
Jash Chandan
Plant Connectivity Management
Manage Notifications by SAP Plant Connectivity
Manage Notifications by PIC
Browse Notifications

Figure 3.11 SAP MII Notification Management

In SAP MII, you can create a notification and publish it to SAP Plant Connectivity. You can create a notification by browsing tags from SAP Plant Connectivity or the PIC. The initial status of a notification in SAP MII is set to **Draft**, but after publishing the notification to SAP Plant Connectivity the status is set to **Draft** in SAP Plant Connectivity. In the **Browse Notifications** page, you can search by SAP Plant Connectivity data server, notification name, and status. Notification details will be displayed in the **Notifications** table, where you can perform activities like export, import, activate, and pause for selected notifications from the table.

3.2 Connecting to External Systems Using Data Servers

Data servers in SAP MII are the most important aspect of manufacturing plant floor connectivity because they provide direct and near-real-time bidirectional access to the manufacturing data present in different systems and applications on the plant floor. Via standard protocols and application interfaces, the data servers provide direct connections to databases or applications in levels 2 and 3 of an enterprise. A considerable list of data connectors is provided in SAP MII. You can configure these as data servers and, using query templates, retrieve or write data to these systems. The list of different connector types available in SAP MII 15.5 is as follows:

- AggregateConnector
- AlarmSuite
- CatalogConnector
- DataSource
- IDBC
- IP21
- IP21OLEDB
- InSQL
- InSQLPCo
- KPIConnector
- MDOConnector
- OLAP
- OLEDB
- OpenConnector
- PCoConnector
- PCoSimulator
- Simulator
- VirtualAlarm
- VirtualCatalog
- VirtualIDBC
- VirtualKPI
- VirtualMDO
- VirtualOLAP
- VirtualPCo
- VirtualTAG
- VirtualXML
- VirtualXacute
- XMLConnector
- XacuteConnector

Note

A few notes on some of these connectors:

- Aggregate Connector was introduced to combine multiple query templates in a single data query. But now this can be achieved more easily and flexibly using BLS transactions.
- AlarmSuite Connector was introduced to connect to Wonderware AlarmSuite, which is currently an obsolete application.
- IP21 Connector was introduced to connect to AspenTech InfoPlus.21, which is a real-time information management system for manufacturing processes. It utilizes the Easysoft JDBC/ODBC driver, which you need to deploy to connect to a system Data Source Name (DSN) on the machine where the InfoPlus.21 suite is installed. Now IP21 Connector is the recommended connector to connect to AspenTech InfoPlus.21, which connects via SAP Plant Connectivity.
- XML Connector was introduced to connect to XML data sources using HTTP, but now it's easier to achieve the same end using BLS transactions. Nevertheless, XML Connector can be used to execute any illuminator service in MII or any HTTP service using XML.

Each data server provides connections to specific applications or systems used on the manufacturing plant floor. Any application or system using one of the previously mentioned connectors can be integrated with SAP MII.

The data servers provide different query modes to retrieve different types of data such as current data, historical data, statistical data, and so on. This data varies with the type of data server and is specified when you're configuring the query templates using the data servers. Also, each mode might have one or more query methods, such as AVG, MIN, MAX, and so on, which provide specific data for that specific query mode.

In the following sections, we'll first look at the configuration of a data server in SAP MII, and then discuss many of the different connector types listed earlier.

3.2.1 Data Server Configuration

You need to create a separate data server configuration in SAP MII for each plant system you are connecting to. You can create the data server configuration from the **Data Servers** menu under the **Data Services** menu category.

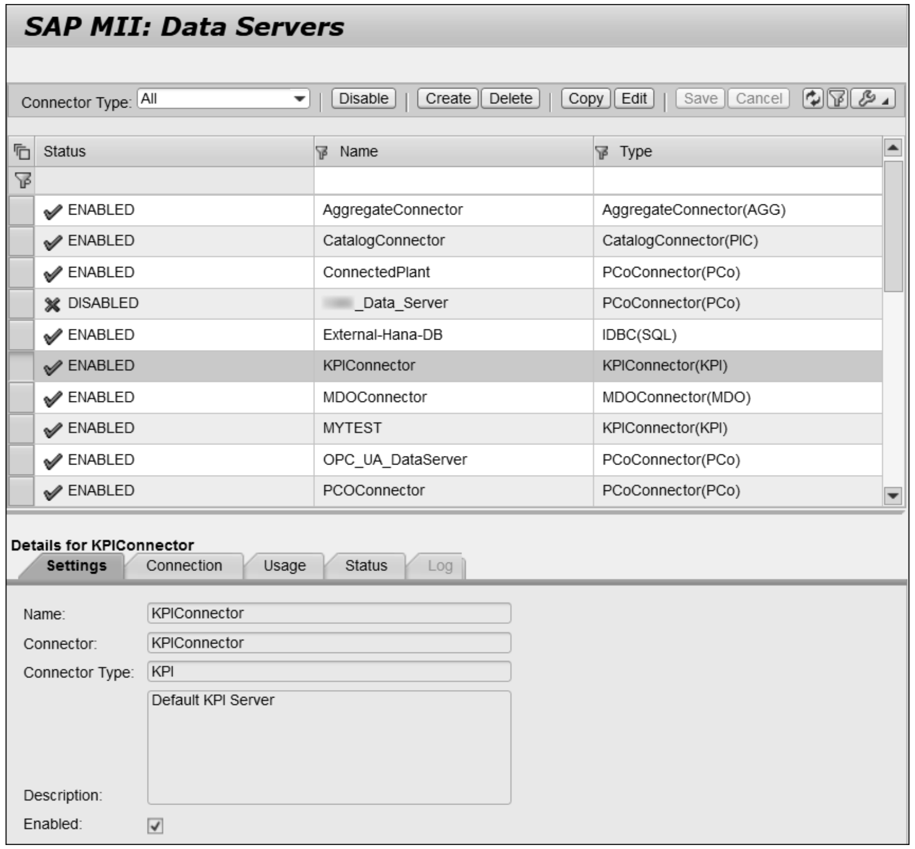
In the **Data Servers** configuration page, shown in Figure 3.12, the existing data servers are shown in a table. The table contains the status, name, and type of each configured data server. You can search for a specific type of data server from the existing data server table by selecting the connector type in the **Connector Type** dropdown field.

To create a new data server configuration, click the **Create** button at the top of table. The new data server configuration screen will appear, where you can enter the server name in **Server Name** field, select a connector type from the **Connector Type** dropdown, and enter a description of a data server in the **Description** field. Next, click the **Finish** button to create your new data server. There are two other buttons available, **Previous** and **Next**, which are enabled based on the type of connector selected. You can copy an existing data server configuration by selecting the existing configuration from the data server table, clicking the **Copy** button, and then entering the name of copied data server. Save the configuration by clicking the **Save** button.

To delete the selected data server configuration, use the **Delete** button from the data server configuration screen. You can edit an existing data server using the **Edit** button and disable a selected data server by clicking the **Disable** button. You can also enable an existing disabled status data server by clicking the **Enable** button. You need to use the **Cancel** button to cancel your changes. You can see the status of all enabled data servers configured in SAP MII from the **Data Services • Connection Status** menu option. The details of a selected connector are displayed at the bottom of the data server table. The details of the connector are displayed in four tabs:

- The **Settings** tab contains the name, connector, connector type, and description information of the data source. It also contains an **Enabled** checkbox, using which you can enable the data server and make it ready to use.
- The **Connections** tab contains connection-related information for the data server. The parameter list in this tab varies depending upon the type of connector.

- The **Usage** tab shows a list of names of objects for which the selected connector is used in SAP MII.
- The **Status** tab displays the status of the selected connector, along with the server name, connector type, used connections, available connections, maximum used connections, and maximum waiting time.



Status	Name	Type
✓ ENABLED	AggregateConnector	AggregateConnector(AGG)
✓ ENABLED	CatalogConnector	CatalogConnector(PIC)
✓ ENABLED	ConnectedPlant	PCoConnector(PCo)
✗ DISABLED	[redacted]_Data_Server	PCoConnector(PCo)
✓ ENABLED	External-Hana-DB	IDBC(SQL)
✓ ENABLED	KPIConnector	KPIConnector(KPI)
✓ ENABLED	MDOConnector	MDOConnector(MDO)
✓ ENABLED	MYTEST	KPIConnector(KPI)
✓ ENABLED	OPC_UA_DataServer	PCoConnector(PCo)
✓ ENABLED	PCOConnector	PCoConnector(PCo)

Details for KPIConnector

Name: KPIConnector
Connector: KPIConnector
Connector Type: KPI
Description: Default KPI Server
Enabled: ☑

Figure 3.12 Data Server Configuration in SAP MII

Max. Retry Count, **Retention Days**, and **Retry Interval** are three connection parameters for the data server that are used for data buffering:

- **Max. Retry Count** indicates the maximum number of times the system will retry execution before stopping.
- The **Retry Interval** value is in milliseconds. It indicates how long the system will wait before retrying.
- **Retention Days** gives the number of days the system will keep the data buffer before removing it.

Note

You can get the status of all configured data servers from the following HTTP service:

http://<host>:<port>/XMII/Illuminator?Service=SystemInfo&Mode=Status

3.2.2 IP21 Connector

You can connect an SAP MII system with an AspenTech InfoPlus.21 data source using the IP21 connector. Before configuring the IP21 connector, you need to install the IP21 ODBC driver and EasySoft JDBC-ODBC bridge. IP21 connector supports different modes, like the IP21OLEDB connector, discussed in the next section.

3.2.3 IP21OLEDB Connector

The IP21OLEDB connector provides connectivity to AspenTech InfoPlus.21, which is a real-time information management system for manufacturing processes. The IP21OLEDB connector uses the SAP MII OLEDB Universal Data Server (UDS) to connect to the AspenTech InfoPlus.21 system.

You can access the data that the InfoPlus.21 suite provides through tag queries, which are created using this data server in the SAP MII workbench. Tag data is time-series data for specific process characteristics, represented by a tag—such as, **Line1Temperature**, **Pump1Flowrate**, and so on. For example, you can retrieve the flow rate of a material or the pressure readings from a production line, which is captured in IP21, in SAP MII by creating a tag query on this data server. You might need this information in various scenarios, such as when you're displaying charts for a process characteristic trend, when doing a quality analysis, when aggregating and updating in SAP ERP or SAP S/4HANA, and so on. The **Mode** settings supported by this connector are **Current**, **HistoryEvent**, **History**, and **Statistics**. It also enables browsing of the IP21 namespace. Because no group hierarchy exists in the IP21 database, the **GroupList** mode does not return any value. In Table 3.9, you can find the general parameters for IP21OLEDB data server configuration.

Property	Description
Message IP	IP address or hostname of the server where the UDS is installed.
Port	The port of the remote machine where the UDS is listening for requests.
TagList	The name of the table that holds a list of all the valid tag names in IP21. Default value is "alltags."

Table 3.9 General Parameters for IP21OLEDB Data Server Configuration

Property	Description
TagListName	The column name of the **TagList** that holds the tag names. Default value is "record_name."
TagListDescription	The column name of the **TagList** that holds the tag description. Default value is "description."
TagInfo	The name of the table from which tag info metadata like description, record type, and range information are fetched for a tag. The default value should be left blank, in which case the name of the tag is used as the table name.
TagInfoName	The column name of the **TagInfo** that holds the tag names. Default value is "name."
TagInfoDescription	The column name of the **TagInfo** that holds the tag description. Default value is "name->description."
TagInfoType	The column name of the **TagInfo** that specifies the IP21 record number. SAP MII interprets #19 and #1649 as discrete data types and #20 and #1640 as analog data types. The rest are interpreted as analog data types. Default value is "definition."
TagInfoMinRange	The column name of the **TagInfo** that provides EGU range information used for scaling charts using server provided ranges. This field provides the minimum range value. Default value is "name->graphminimum."
TagInfoMaxRange	The column name of the **TagInfoTable** that provides EGU range information used for scaling charts using server provided ranges. This field provides the maximum range value. Default value is "name->graphmaximum."
Writable	Specifies whether or not you can update current values to a tag using the TagQuery.

Table 3.9 General Parameters for IP21OLEDB Data Server Configuration (Cont.)

Apart from the general properties just explained, you need to configure the query mode–specific parameters as explained in the following sections. If a specific query mode is not used for a data server, then that mode-specific parameter need not be configured.

History Mode Parameters

History mode enables you to retrieve historical time-series data in an interpolated manner available in the tag data source. You need to configure the parameters shown in Table 3.10 for **History** mode. All parameters explained in this table are mandatory for using **History** mode.

Property	Description
HistoryTable	The name of the table from where time interpolated history data should be retrieved. Default value should be the name of the history table, which is commonly named "HISTORY."
HistoryName	Specifies the column name in the **HistoryTable** that represents the tag name. Default value is "NAME."
HistoryTime	Specifies the column name in the **HistoryTable** that represents the tag value's timestamp. Default value is "TS."
HistoryValue	Specifies the column name in the **HistoryTable** that represents the tag value at a specific timestamp. Default value is "VALUE."
HistoryQuality	Specifies the column name in the **HistoryTable** that represents the quality value of the tag data. Default value is "QUALITY."
HistoryPeriod	Specifies the column name in the **HistoryTable** that sets the data retrieval period. Default value is "PERIOD."

Table 3.10 History Mode Parameters for IP21OLEDB Data Server Configurations

HistoryEvent Mode Parameters

The **HistoryEvent** mode enables you to retrieve historical event data available in the tag data source. The **HistoryEvent** mode supports two query methods: normal and compressed.

The *normal* method is invoked when the value of the **Method** field in the query template is blank. The connector then uses the values of the table and column names in the fields of **HistoryEventName**, **HistoryEventTime**, **HistoryEventValue**, and **HistoryEventQuality** in the connector configuration to retrieve the data. The *compressed* method is not available in the method dropdown and has to be entered manually. This method uses the table and column names in the fields of **CompressedHistoryEventName**, **CompressedHistoryEventTime**, **CompressedHistoryEventValue**, and **CompressedHistoryEventQuality** to retrieve the data. These two methods are relevant when you need to maintain and access data with different granularity. The compressed data storing method in the historian does not store all data points but uses different algorithms to reject data that can be interpreted at runtime. You need to configure the parameters in Table 3.11 for **HistoryEvent** mode.

Property	Description
HistoryEventTable	Specifies the name of the table from where history data should be retrieved. Default value is blank, in which case the name of the tag is used as the table name.
HistoryEventName	Specifies the column name from the **HistoryEventTable** that represents the tag name. Default value is "NAME."
HistoryEventTime	Specifies the column name from the **HistoryEventTable** that represents the tag value's timestamp. Default value is "IP_TREND_TIME."
HistoryEventValue	Specifies the column name from the **HistoryEventTable** that represents the tag value. Default value is "IP_TREND_VALUE."
HistoryEventQuality	Specifies the column name in the **HistoryEventTable** that represents the quality value of the tag data. Default value is "IP_VALUE_QSTATUS."
CompressedHistoryEventTable	Specifies the name of the table from where history data should be retrieved in case the method used in the query is "Compressed." Default value should be left blank, in which case the name of the tag is used as the table name.
CompressedHistoryEventName	Specifies the column name from the **CompressedHistoryEventTable** that represents the tag name. Default value is "NAME."
CompressedHistoryEventTime	Specifies the column name from the **CompressedHistoryEventTable** that represents the tag value's timestamp. Default value is "IP_TREND_TIME."
CompressedHistoryEventValue	Specifies the column name from the **CompressedHistoryEventTable** that represents the tag value. Default value is "IP_TREND_VALUE."
CompressedHistoryEventQuality	Specifies the column name in the **CompressedHistoryEventTable** that represents the quality value of the tag data. Default value is "IP_VALUE_QSTATUS."

Table 3.11 History Event Mode Parameters for IP21OLEDB Data Server Configurations

Current Mode Parameters

Current mode enables you to retrieve tag values at the current timestamp. You need to configure the parameters in Table 3.12 for the tag query using **Current** mode.

Property	Description
CurrentTable	The name of the table from where the current value of the tag data should be retrieved. Default value is blank, in which case the name of the tag is used as the table name.
CurrentNameColumn	Specifies the column name in the **CurrentTable** that represents the tag name. Default value is "NAME."
CurrentValueColumn	Specifies the column name in the **CurrentTable** that represents the tag value. Default value is "IP_VALUE."
CurrentTime	Specifies the column name in the **CurrentTable** that represents the timestamp of the selected tag. Default value is "IP_VALUE_TIME."
CurrentQuality	Specifies the column name in the **CurrentTable** that represents the quality value of the tag data. Default value is "IP_VALUE_QUALITY."

Table 3.12 Current Mode Parameters for IP21OLEDB Data Server Configurations

You need to configure an instance for the IP21 system in the UDS of type OLE DB UDS and point it to the server where the AspenTech InfoPlus.21 system is installed.

3.2.4 InSQL Connector

The InSQL connector provides the connectivity to Wonderware IndustrialSQL Server (InSQL). Using this data server, you can create tag queries from the SAP MII workbench to retrieve process time-series data such as machine readings, temperature, and line speed for a production line as current data or historical data for a time range. The InSQL connector is a special type of IDBC connector because it connects to a JDBC-compliant database provided by InSQL, so most of its configuration parameters are the same as IDBC's parameters. Set the **Use OLEDB** property to **True** for InSQL version 8.x and later, and **False** for InSQL version 7.x and earlier. You use **ServerURL** to specify the connection string to the InSQL JDBC-compliant database, and you specify corresponding authentication details in the **UserName** and **Password** properties.

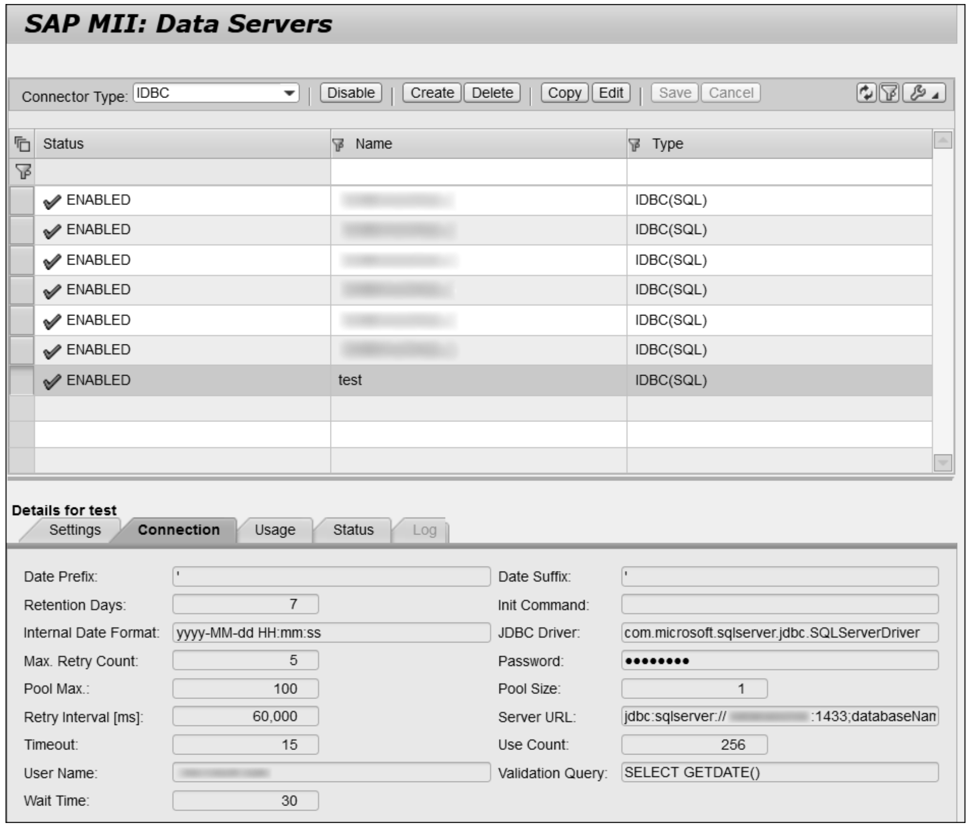
3.2.5 InSQLPCo Connector

The InSQLPCo connector is similar to the InSQL connector except for the type PCo that can be used in SAP Plant Connectivity and the PIC query as a source data server.

3.2.6 IDBC Connector

The IDBC connector is the most commonly used connector in SAP MII and provides connectivity to legacy databases supporting Java Database Connectivity (JDBC). This connector provides direct connection to any relational database to create a SQL query on its table data. Examples include Microsoft SQL Server, Oracle Database, SAP MaxDB, DB2, SAP HANA database, and so on. You can also use this connector to connect to any application database, such as an MS SQL Server database of an MES. This is useful when you want to retrieve or update data from the composite application to a custom legacy database or any other application database. You need to change the following properties for authentication while creating a new IDBC data server configuration, as shown in Figure 3.13:

- **UserName**
- **Password**
- **ServerURL**



Connector Type: IDBC

Status	Name	Type
ENABLED	[redacted]	IDBC(SQL)
ENABLED	[redacted]	IDBC(SQL)
ENABLED	[redacted]	IDBC(SQL)
ENABLED	[redacted]	IDBC(SQL)
ENABLED	[redacted]	IDBC(SQL)
ENABLED	[redacted]	IDBC(SQL)
ENABLED	test	IDBC(SQL)

Details for test

Date Prefix:	'	Date Suffix:	'
Retention Days:	7	Init Command:	
Internal Date Format:	yyyy-MM-dd HH:mm:ss	JDBC Driver:	com.microsoft.sqlserver.jdbc.SQLServerDriver
Max. Retry Count:	5	Password:	••••••••
Pool Max.:	100	Pool Size:	1
Retry Interval [ms]:	60,000	Server URL:	jdbc:sqlserver://[redacted]:1433;databaseNan
Timeout:	15	Use Count:	256
User Name:	[redacted]	Validation Query:	SELECT GETDATE()
Wait Time:	30		

Figure 3.13 IDBC Data Server Configurations

The **ServerURL** parameter specifies the database connection string that is different for different databases. For example, for MS SQL Server it would be as follows:

jdbc:sqlserver://<host>:<port>;databaseName=<dbName>

In addition, you need to upload and deploy a JDBC driver from the **System Resources • JDBC Drivers** menu. You also need to specify the database driver package name in the **JDBCDriver** property according to the database used (e.g., com.inet.tds.TdsDriver for MS Server 8.0 or com.microsoft.sqlserver.jdbc.SQLServerDriver for MS SQL Server). Other properties available in the IDBC data server configuration are as follows:

- The **Init Command** field value is used by the connector executed when a connection pool is established.
- The **Validation Query** field is used to check if the database connection is valid. For example, the `SELECT GETDATE()` query is used for Microsoft SQL Server.
- The **Pool Max** field value indicates the maximum number of connections allocated by the pool.
- The **Pool Size** field value specifies the number of connections that will be established during the initialization of the connection pool.
- The **Use Count** field value indicates the number of times a connection is used before it's terminated, releasing the resource.
- The **Wait Time** field value indicates the amount of time the SAP MII system will wait for a connection before the connection is failed.

You can use the default values provided with the new configuration for other required properties.

3.2.7 OLAP Connector

The OLAP connector provides connectivity to multidimensional data sources such as online analytical processing (OLAP)–based systems supporting the XML for Analysis (XMLA) specification. Examples of OLAP data sources include SAP BusinessObjects Business Intelligence (SAP BusinessObjects BI) and MS SQL Server Analysis Services. The key elements of an OLAP data source are InfoCubes, multidimensional data structures used for faster execution of complex analytical queries. You can use this data server to create a Multidimensional Expressions (MDX) query over an InfoCube or XMLA data source, the output of which you can use in analysis charts and reports. Some examples might be product yield reports sorted by work centers and by months. The configuration properties in Table 3.13, which are specific to the OLAP data server, are available in the OLAP connector configuration.

Property	Description
Catalog	Specifies the XMLA server or database instance name. For SAP BusinessObjects BI, it specifies the InfoCube name.
ColumnNameAttribute	Specifies the string name used by the XMLA server implementation to retrieve the returned column label. For MS SQL Server Analysis Services, the default value is "sql:field." For SAP BusinessObjects BI, the default value is "name."
DataSourceInfo	Specifies the Microsoft parameter for connecting to a data store (e.g., Provider=MSOLAP; Data Source=local). Not required for SAP BusinessObjects BI.
DataTypeAttribute	Specifies the string in XMLA used to return the column data types in the response document. Default value is "type."
DecimalDelimiter	Specifies the character used for decimal notation. It can be either a period (.) or a comma (,), as configured in the OLAP system.
IP	Specifies the IP address of the server hosting the OLAP services.
Port	Specifies the listening port of the XMLA service on the OLAP server.
ResultInNamespace	Specifies the Boolean indicator for returning a SOAP response from the XMLA server. Specify **True** for SAP BusinessObjects BI.
WebService	Specifies the entry point to the XMLA service in the OLAP server. Specify "/sap/bw/xml/soap/xmla" for SAP BusinessObjects BI and "/xmla/msxisapi.dll" for MS SQL Server Analysis Services. You need to enable the XMLA web interface in SAP BusinessObjects BI for using the OLAP data server.

Table 3.13 OLAP Data Server Configuration Properties

Note

The OLAP data server status doesn't display any number of available or used connections. This is because OLAP is an on-demand connection, not a persistent connection, meaning that the connection is established not by a connection pool, but only when required to execute a query.

3.2.8 Open Connector

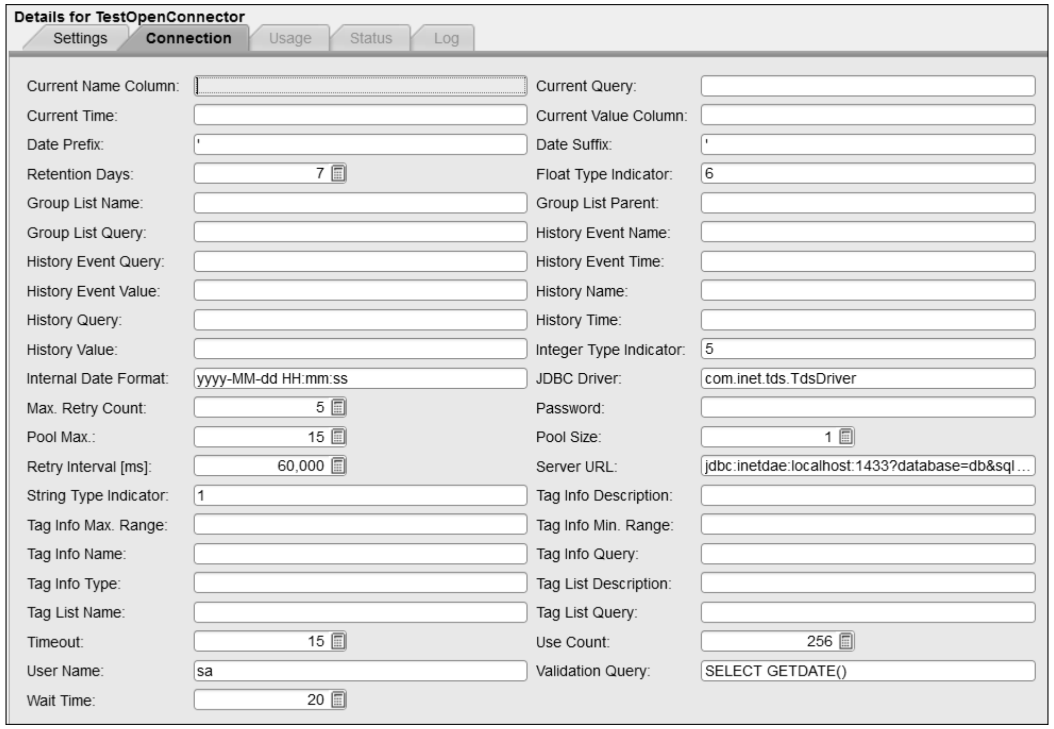
You can use the open connector to connect to tag-based relational databases to retrieve historical data based on time periods. Using this connector, you can connect

SAP MII to process historian databases, which store process characteristic data in a time-series fashion and provide the following type of tag-based data:

- Current values
- Historical (time-series) values
- Statistical summaries of historical data

In contrast to the previous query modes, the open connector also supports querying tag groups and tag names from the tag data source using the **GroupList** and **TagList** modes. You can create a tag query in SAP MII using this data server to retrieve current or historical process characteristic information such as temperature readings and flow rates. You can use this data server for a custom process historian using any relational database such as Oracle, MS SQL Server, or DB2 from which you can query time-series data. It also acts as a tag data source. The **TagListQuery**, **CurrentQuery**, **HistoryQuery**, and **HistoryEventQuery** parameters should contain the SQL query to select the tag data.

You can use this connector only to query the data from the tag database; you can't write data back to it. Because the open connector uses JDBC data sources, the connection properties such as **JDBCDriver**, **ServerURL**, **UserName**, and **Password** are the same as for the IDBC connector. Apart from these properties, the open connector has certain specific parameters in the data server configuration, as shown in Figure 3.14.



Details for TestOpenConnector

Current Name Column:		Current Query:	
Current Time:		Current Value Column:	
Date Prefix:	'	Date Suffix:	'
Retention Days:	7	Float Type Indicator:	6
Group List Name:		Group List Parent:	
Group List Query:		History Event Name:	
History Event Query:		History Event Time:	
History Event Value:		History Name:	
History Query:		History Time:	
History Value:		Integer Type Indicator:	5
Internal Date Format:	yyyy-MM-dd HH:mm:ss	JDBC Driver:	com.inet.tds.TdsDriver
Max. Retry Count:	5	Password:	
Pool Max.:	15	Pool Size:	1
Retry Interval [ms]:	60,000	Server URL:	jdbc:inetdae:localhost:1433?database=db&sql ...
String Type Indicator:	1	Tag Info Description:	
Tag Info Max. Range:		Tag Info Min. Range:	
Tag Info Name:		Tag Info Query:	
Tag Info Type:		Tag List Description:	
Tag List Name:		Tag List Query:	
Timeout:	15	Use Count:	256
User Name:	sa	Validation Query:	SELECT GETDATE()
Wait Time:	20		

Figure 3.14 Open Connector Data Server Configurations

However, not all these parameters are mandatory; it depends on which query modes of the open connector the data server instance you want to use supports. The following sections explain the parameters required for different query modes.

GroupList Mode Parameters

GroupList mode enables retrieving information about tag groups available in the tag data source. You need to configure the parameters in Table 3.14.

Property	Description
GroupListName	Specifies the column name that returns the tag group name in the **GroupList** query. Default value is "Name."
GroupListParent	Specifies the column name from the **GroupList** query that returns the group's parent, if applicable.
GroupListQuery	Specifies the SQL query string that returns the list of tag groups available within the data source. It should include and match the **GroupListName** and **GroupListParent** as specified in the corresponding properties, as required.

Table 3.14 GroupList Mode Parameters for Open Connector Data Server Configuration

TagList Mode Parameters

TagList mode enables you to retrieve tag information and metadata from the tag data source. To use this mode, you need to configure the parameters in Table 3.15.

Property	Description
FloatTypeIndicator	Specifies an integer to indicate the tag data type. The floating-point data type is specified as 6, which is a default setting and should not be changed.
IntegerTypeIndicator	Specifies an integer to indicate the tag data type. The integer data type is specified as 5, which is a default setting and should not be changed.
StringTypeIndicator	Specifies an integer to indicate the tag data type. The string data type is specified as 12, which is a default setting and should not be changed.
TagInfoName	Specifies the column from the **TagInfoQuery** that represents the tag name.
TagInfoDescription	Species the column from the **TagInfoQuery** that represents the tag description.

Table 3.15 TagList Mode Parameters for Open Connector Data Server Configurations

Property	Description
TagInfoMinRange	Species the column from the **TagInfoQuery** that represents the minimum engineering range.
TagInfoMaxRange	Species the column from the **TagInfoQuery** that represents the maximum engineering range.
TagInfoType	Species the column from the **TagInfoQuery** that represents the tag's data type.
TagInfoQuery	Species the SQL query string that returns the tag information.
TagListName	Species the column from the **TagList** query that represents the tag name.
TagListDescription	Species the column from the **TagList** query that represents the tag description.
TagListQuery	Specifies the SQL query string that returns the tag list.

Table 3.15 TagList Mode Parameters for Open Connector Data Server Configurations (Cont.)

History Mode Parameters

History mode enables you to retrieve historical time-series data in an interpolated manner available in the tag data source. You need to configure the parameters in Table 3.16 for this mode.

Property	Description
HistoryName	Specifies the column name in the **HistoryQuery** that represents the tag name.
HistoryTime	Specifies the column name in the **HistoryQuery** that represents the tag value's timestamp.
HistoryValue	Specifies the column name in the **HistoryQuery** that represents the tag value at a specific timestamp.
HistoryQuery	Specifies the SQL query string that returns the historical data from the tag data source.

Table 3.16 History Mode Parameters for Open Connector Data Server Configurations

Current Mode Parameters

Current mode enables you to retrieve tag values at the current timestamp. You need to configure the parameters in Table 3.17 for this mode.

Property	Description
CurrentNameColumn	Specifies the column name in the **CurrentQuery** that represents the tag name.
CurrentValueColumn	Specifies the column name in the **CurrentQuery** that represents the tag value.
CurrentTime	Specifies the column name in the **CurrentQuery** that represents the timestamp of the selected tag.
CurrentQuery	Specifies the SQL query string that returns the current tag values from the data source.

Table 3.17 Current Mode Parameters for Open Connector Data Server Configurations

3.2.9 OLEDB Connector

OLEDB is a Microsoft API to enable access to data sources using the Microsoft component object model (COM). The SAP MII OLEDB connector provides access to OLEDB-compliant data sources via SAP Plant Connectivity. You can use this data server to connect to MS Access, MS Excel files, OSIsoft PI, and GE Fanuc Proficy Historian and create SQL queries on its data. This capability is useful when you need to retrieve data from or write data into an Excel file that might be generated by or be sent to another plant system. You need to configure the connection parameters as shown in Table 3.18 to create an OLEDB connector in SAP MII.

Field Name	Purpose
Date Prefix	Prefix used for **DateTime** database columns.
Date Suffix	Suffix used for **DateTime** database columns.
Internal Date Format	The date format.
Max. Retry Count	Used for data buffering. Indicates the maximum number of times the system will retry execution before stopping.
Retry Interval [ms]	Used for data buffering. The value is in milliseconds. It indicates the amount of time the system will wait before retrying.
IP	The hostname or the IP address of the source system.
Port	The port of the remote system where the OLEDB is listening for requests.
Query Timeout [sec.]	Timeout value of the query in seconds.
Writable	If checked, provides write access to the tags on the data source.

Table 3.18 Connection Parameters of OLEDB Connector

This is a type of PCoConnector, which needs the OLEDB source system and agent in SAP Plant Connectivity. In SAP Plant Connectivity, you need to create the source using the OLE DB source system and you need to create an agent for same source system. You can configure the agent in SAP MII using PCoConnector and access the data source.

3.2.10 Simulator Connector

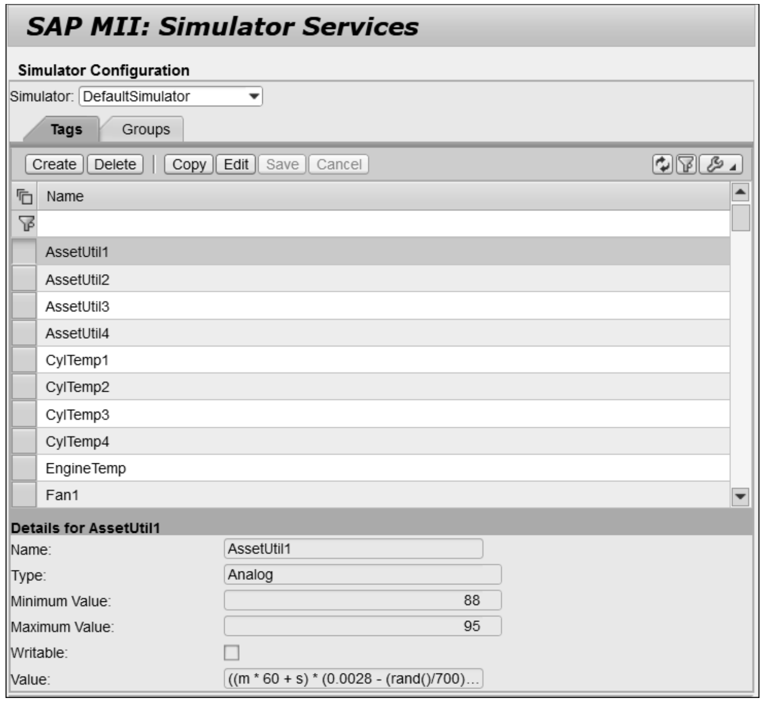
You can use the simulator connector to connect to a simulated tag data source available in SAP MII. You can use this for demonstrating, testing, or debugging purposes. The simulator connector is configured with an XML file that defines groups, tags, and their values. The XML file can be modified using the **Simulator Services** screen. It supports the following modes:

- **Current**
- **CurrentWrite**
- **History**
- **HistoryEvent**
- **ModeList**
- **Statistics**
- **TagList**

Using this data server, you can create PCoQueries based on the simulated XML-based tag data source. You can use the default simulator data server available or create new ones as required with different names. In the settings, you can only change the **Description**. You can configure the tags available for a simulator data server using the **Simulator** field (which shows available simulator instance name) under **Simulator Services**.

To change the simulator tags for the default connector or to create the simulator tags for a new simulator data server, open the **Simulator Services** page from the menu **Data Services • Simulator Services**. In the **Simulator Tags** tab, select the simulator data server instance name from the **Simulator Instance** dropdown. The tags for the default simulator are already available in SAP MII, as shown in Figure 3.15. Select an existing tag or create a tag by clicking the **Create** button.

In the **Name** property, specify a unique name for the tag. In the **Type** property, the choices are **Analog**, **Discrete**, and **String**. Use **Analog** for decimal values, such as 192.743; **Discrete** returns 1 or 0; and **String** returns alphanumeric values (e.g., X25, D38, etc.). The **Minimum Value** and **Maximum Value** specify the range of the returned value of the tag. If you mark a tag as **Writable**, then it enables you to write a new tag value using the tag queries. In the **Value** property, specify a formula to return a tag value. You can use time and mathematical expressions to specify the formula. After creating one or more tags, you need to add those to simulator groups from the **Simulator Groups** tab.



SAP MII: Simulator Services

Simulator Configuration

Simulator: DefaultSimulator

Name
AssetUtil1
AssetUtil2
AssetUtil3
AssetUtil4
CylTemp1
CylTemp2
CylTemp3
CylTemp4
EngineTemp
Fan1

Details for AssetUtil1

Name: AssetUtil1
Type: Analog
Minimum Value: 88
Maximum Value: 95
Writable: □
Value: ((m * 60 + s) * (0.0028 - (rand()/700)…

Figure 3.15 Simulator Services

Note:

You can use the following characters in the expression for dynamic replacement:

- h: Current hour
- m: Current minute
- s: Current second
- r: Current row number in the result set
- n: Total number of rows in the result set

You can also use mathematical functions such as rand(), sin(), cos(), log(), and so on in the expressions for mathematical calculations.

Example expressions are as follows:

- *((m × 60 + s) × (0.0097 + (rand() ÷ 100))) + 815*
- *250 + 250 × sin((r % 60) × pi ÷ 30)*

3.2.11 Virtual Servers

In an SAP MII implementation scenario with multiple plants, it's generally advisable to have one instance of SAP MII running per plant, with a separate instance at the corporate level. It's ideally assumed that the plant-level SAP MII server instances will talk and exchange data among themselves and the central instance, which would mainly serve the purpose of an analytics and reporting dashboard.

You can use virtual server connectors for integration scenarios that involve exchanging data by accessing the data servers of SAP MII server instances residing in different networks and time zones. Data servers not defined in a specific SAP MII server instance's network can still be accessed using virtual server connectors via the SAP MII server on which they're defined as data servers. The data server connector also provides an option for setting an offset to connect to servers in different time zones. This enables the retrieval of corresponding data from the target SAP MII server without changing the existing queries.

There are 10 types of virtual server connectors in SAP MII 15.5: VirtualAlarm, VirtualCatalog, VirtualIDBC, VirtualKPI, VirtualMDO, VirtualOLAP, VirtualPCo, VirtualTAG, VirtualXML, and VirtualXacute. These correspond respectively to their Alarm, Catalog, IDBC, KPI, MDO, OLAP, PCo, TAG, XML, and Xacute connectors. Therefore, using the virtual connectors allows you not only to connect to data server types such as IDBC, OLAP, or any tag-based data source, but also to invoke BLS transactions from another SAP MII installation via the VirtualXacute connector. Using virtual connectors is faster in performance than the usual SOAP or HTTP calls as they use binary mode for data transfer.

To connect to the SAP MII data server using the virtual connector, you need to know the IP or hostname of the server, the username and password of an SAP MII user who has appropriate rights to use the connector, and the name of the data server you're connecting to.

We'll now use an example to demonstrate how interconnectivity can be achieved using the SAP MII virtual server connector. In this example, we'll try to connect to a SQL data server called PlasticResin, which has already been defined in the target SAP MII system using the IDBC connector. The first step is to define a virtual data server in the source SAP MII server to connect with this data server on the target system.

On the **Data Servers Configuration** page, click **Create** and select **VirtualIDBC** as the connector. Then click the **Next** button and select the **Enter Target SAP MII Server Properties Manually** option. Next, click the **Finish** button. On the **Connection** tab, enter the configuration parameters as explained in Table 3.19.

Property	Description
DaysRetention	The number of days the entries of a buffered query should be retained in the buffer.
IP	The IP address/machine name of the remote SAP MII server.
LegacyURL	Check this box to enable support for older SAP MII servers.

Table 3.19 Virtual Server Configuration Properties

Property	Description
MaxRetryCount	Number of times to retry a connection if an attempt fails.
Port	The port to which the remote SAP MII server is listening.
Protocol	Default setting is "http"; however, this can be set to "https" if SSL has been set up.
RemoteLogonName	The login name for the remote SAP MII server (not the remote data server) with proper access rights to the remote data source you want to access.
RemoteLogonPassword	The password for the remote SAP MII server.
RemoteServerName	The name of the data server that you want to access.
RetryInterval	Interval between retry attempts, in milliseconds.
TZ Offset	Describes the time difference between the local and remote SAP MII servers. The time value is measured in minutes.
Destination Name	SAP NetWeaver destination name.
Maintain Session	Used for session connection to avoid multiple logins.
Use Destination Service	Enable when an SAP NetWeaver destination is used and the destination name is provided in the **Destination Name** field.

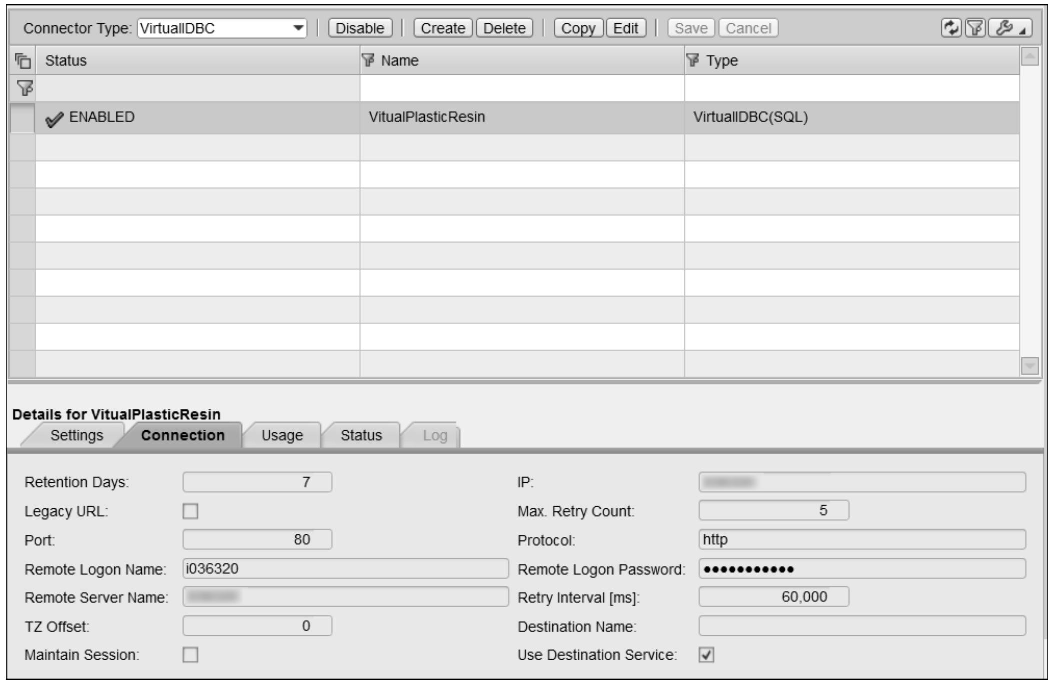
Table 3.19 Virtual Server Configuration Properties (Cont.)

After entering the connection parameters, you can enable the server using the **Enable** checkbox in the **Settings** tab. Next, click **Save** to create a new virtual server called VitualPlasticResin. This connects to the PlasticResin data source in the target system, as shown in Figure 3.16.

The last step is to check whether the connection is actually working. Click the **Status** tab to check the server status.

You can now create data queries using the data server, which refers to another data server in a remote SAP MII installation. Launch the SAP MII workbench and create a new SQL query template as explained in Chapter 4. In the data source view, VitualPlasticResin should show up as an available data source. Select **VitualPlasticResin** as the **Data Source** and set **Mode** to **Query**.

Shift to the **SQL Query Details** configuration page. If a successful connection was made, all the tables from the remote data server should show up under the **Available Tables** list.



Status	Name	Type
✓ ENABLED	VitualPlasticResin	VirtualIDBC(SQL)

Details for VitualPlasticResin

Retention Days:	7	IP:	[redacted]
Legacy URL:	□	Max. Retry Count:	5
Port:	80	Protocol:	http
Remote Logon Name:	i036320	Remote Logon Password:	●●●●●●●●●●●
Remote Server Name:	[redacted]	Retry Interval [ms]:	60,000
TZ Offset:	0	Destination Name:	
Maintain Session:	□	Use Destination Service:	☑

Figure 3.16 Virtual Data Server Configuration

Troubleshooting

If the connection to the virtual server fails:

- Check whether the username and password you have entered are correct. You should specify the username and password of the remote SAP MII server, not of the remote data source.
- Check whether the target SAP MII server is accessible over the network.
- If you're accessing the remote server through a proxy server, verify whether the proxy server is configured under **System Management • System Properties**.

3.2.12 XML Connector

The XML connector is used to connect with an XML-compliant data source to query data from SAP MII. The SAP MII system provides a default XML connector, but you can add other XML connectors. There is nothing to configure on the default XML connector. You need to create XML queries using the XML connector to fetch data from any data source supporting HTTP protocol in XML format. The data query returns an XML document, and the XML data file includes three levels, like the following:

- Rowsets
  - Rowset
    - Columns
    - Rows

Rowsets contains attributes like the document creation date, software version, and start date and end date settings used by the query.

3.2.13 Xacute Connector

The Xacute connector is an internal system connector for SAP MII that you can use to execute BLS transactions as data queries (called Xacute queries) created using the workbench, which is explained in Chapter 6. You always have a default Xacute data server configuration named XacuteConnector available along with the standard installation of SAP MII. You can use this default configuration for creating and executing Xacute queries. Only the **Description** and **AutoBind** properties are modifiable in the Xacute data server configuration. If the **AutoBind** checkbox is checked, then the SAP MII session variables are automatically mapped to the transaction that has transaction properties with the same name. This is explained in detail with examples in Appendix A, Section A.6.

3.2.14 PCo Connector

In SAP MII, the PCo connector is used to connect with an SAP Plant Connectivity instance to perform query functionality. To create the data server of type PCoConnector, you need to go to the SAP MII menu under **Data Servers** and click the **Create** button. The system will open a new page where you can provide the **Server Name** and **Description** and can select **PCoConnector** as the **Connector Type**. Now click the **Next** button and the system will ask you to provide the URL of the SAP Plant Connectivity instance. Either you can enter the URL of the SAP Plant Connectivity management web service manually, or you can browse for it from the System Landscape Directory (SLD). The system will ask you for your user ID and password for the SAP Plant Connectivity management web service interface if required. In the **Connection** tab, you can find the properties described in Table 3.20.

Connection Parameter	Parameter Description
Agent Name	Name of the SAP Plant Connectivity agent created in the SAP Plant Connectivity system.
Agent IP	IP address of the system where the agent is created and SAP Plant Connectivity is installed.

Table 3.20 PCo Connector Connection Properties

Connection Parameter	Parameter Description
Agent Port	Port of the agent configured in SAP Plant Connectivity.
WebSocket Port	You can specify the SAP Plant Connectivity port in the **WebSocket Port** field; it is used when you use SAP Plant Connectivity as a WebSocket server (e.g., from the UI).
Certificate Keystore	Used from the keystore in SAP NetWeaver Administrator if you want a secure (HTTPS) connection between SAP MII and SAP Plant Connectivity. Used for certificate-based authentication using SSL.
Certificate Name	This names the SSL certificate used for encryption when the SSL connection is used.
Date Prefix	It refers prefix to the date string where date values are compared.
Date Suffix	It refers suffix to the date string where date values are compared.
Internal Date Format	The format used to construct `WHERE` clause conditions.
Retention Days	The number of days the query remains in SAP MII before being removed.
Max. Retry Count	The number of attempts used to establish the connection.
PCo Mgmt Service URL	The URL of the SAP Plant Connectivity management service (*http://<server>:<port>/PCoManagement?wsdl*).
PCo Mgmt Service User	The username used to access the SAP Plant Connectivity management web service interface.
PCo Mgmt Service Password	The password for the SAP Plant Connectivity management service user.
Query Timeout [sec]	The time in seconds before timeout of a query.
Retry Interval [ms]	The time in milliseconds after which the next connection attempt should be made.
Use SSL	Specifies whether a secured connection will be made between SAP Plant Connectivity and SAP MII via SSL.
Writable	If you enable this checkbox, the system will allow you to run PCoQuery in **TagStoreQuery** mode so that you can add tag values.

Table 3.20 PCo Connector Connection Properties (Cont.)

Note

Before configuring the PCoConnector type data server in SAP MII, you need to install the SAP Plant Connectivity system and perform the following activities in SAP Plant Connectivity:

1. Create a source system in SAP Plant Connectivity.
2. Create an agent using the created source system.
3. Enable the **Run Agent Instance as Executable** checkbox (optional).
4. Configure the query ports.

3.2.15 SAP MII and SAP HANA Integration Using SAP HANA Smart Data Integration

The SAP HANA smart data integration adapter helps you get shop floor machine or sensor data into the SAP HANA system virtually without copying data to the SAP HANA system. Using the SAP HANA smart data integration adapter, you can connect to the SAP MII system where the data from the SAP Plant Connectivity queries will be available as virtual tables in SAP HANA. Next, you can use the virtual tables' data for real-time analysis in SAP HANA. To configure the SAP HANA smart data integration adapter and access data from SAP MII, you need to perform the following steps (these steps assume that SAP HANA Studio is available in your local system):

1. Open the **Tools and Archives** page (open *http://<Host>:<Port>/XMII/Menu.jsp* in your browser, then follow menu path **Content Development • Tools and Archives**) in SAP MII to download the SAP MII SDI adapter.
2. After successful download, extract the *MII_SDI_Adapter_1.0.1.jar* file to your local system.
3. Deploy the *MII_SDI_Adapter_1.0.1.jar* file using the SAP HANA Data Provisioning Agent configuration tool.
4. Open SAP HANA Studio and expand the **HANA** system node. To create a remote source system, follow menu path **Provisioning • Remote Sources**, then right click and select **New Remote Source**, as shown in Figure 3.17. Enter the values for the required fields, such as **Source Name** (you can provide any name), **Adapter Name** (select **MII_SDI_Adapter**), **Source Location** (select the source agent by which the adapter is deployed), **Server Name** (enter the SAP MII server name), **Port** (provide the SAP MII system port), **SSL Mode** (choose **Enabled** or **Disabled**), **String Column Length** (specify the column length of all the string columns), **User Name**, and **Password** (provide the user name and password for connecting to the SAP MII system).
5. Save the new connection by clicking **Save**.
6. You can view all the SAP MII projects by expanding the **Remote Source System** node.

7. To add the virtual tables, you can select any PCo query from the project, then right-click and select **Add as Virtual Table**. Next you can select the schema and select **Create**. After that, virtual tables will be added to the selected schema.

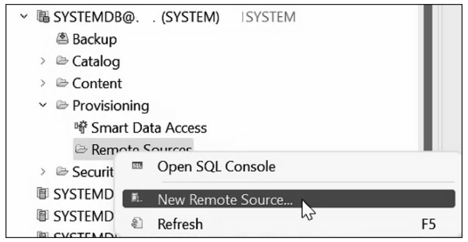
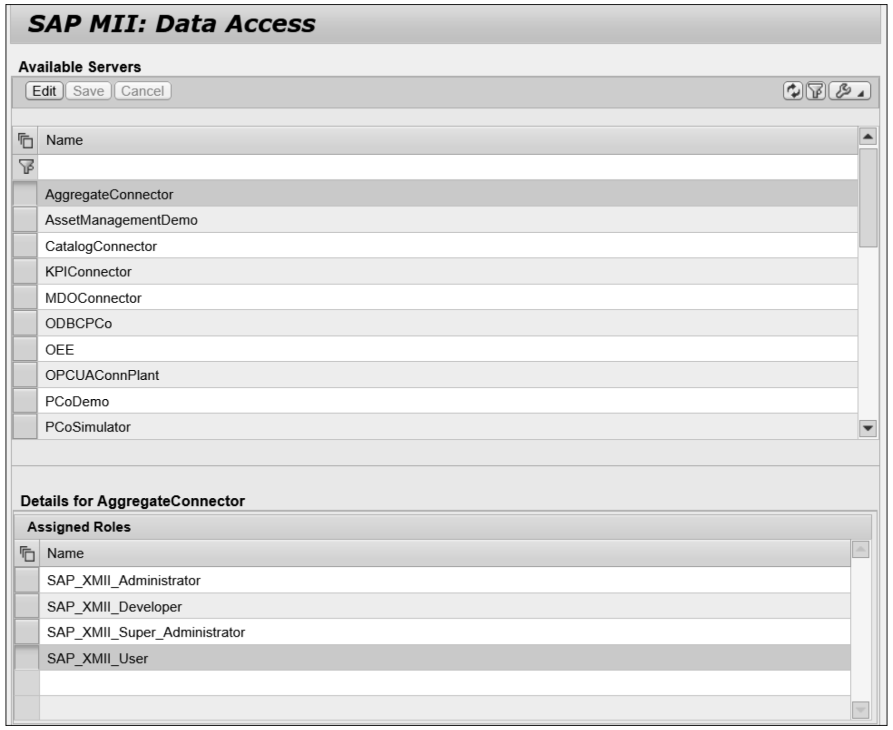


Figure 3.17 SAP MII Integration with SAP HANA Studio

3.3 Data Access

The **SAP MII: Data Access** screen, shown in Figure 3.18, enables the administrator to assign SAP MII user roles to the data servers defined using the **Data Servers** screen (explained in Section 3.2) under the **Data Services** menu category. Without the roles assigned to the data server, no data query can be created or executed by the user using that specific data server.



Name
AggregateConnector
AssetManagementDemo
CatalogConnector
KPIConnector
MDOConnector
ODBCPCo
OEE
OPCUAConnPlant
PCoDemo
PCoSimulator

Details for AggregateConnector

Assigned Roles

Name
SAP_XMII_Administrator
SAP_XMII_Developer
SAP_XMII_Super_Administrator
SAP_XMII_User

Figure 3.18 Configuring Data Access

To assign roles to a data server, you need to select the data server from the **Available Servers** list and click the **Edit** button. Two tables, **Available Roles** and **Assigned Roles**, will be displayed at the bottom of the **Available Servers** list table. You can search for roles in the **Available Roles** table, then assign the role to the **Assigned Roles** table using the **Add** or **Add All** button. As per your requirements, you can also remove roles from the **Assigned Roles** table by using the **Remove** or **Remove All** button. When the assignment or removal is complete, click the **Save** button.

3.4 Message Services: Synchronizing the Shop Floor with the Enterprise

Data captured and generated in enterprise systems (e.g., SAP ERP or SAP S/4HANA), such as production orders, material inventory data, and work center capacity information, might need to be synchronized with the manufacturing shop floor for detailed execution planning and actual execution. Shop floor data synchronization can be done through SAP MII, where shop floor manufacturing-related master data as well as order data are downloaded from SAP ERP or SAP S/4HANA to SAP MII and shop floor manufacturing confirmation data are sent from SAP MII to SAP ERP or SAP S/4HANA. To synchronize data between SAP ERP or SAP S/4HANA and the shop floor, you need to perform required activities that are discussed in the following sections. In Section 3.4.1, you'll learn about synchronizations and SAP MII message services. You'll learn about the required configurations for message listeners in SAP MII in Section 3.4.2 and about message processing rules configuration in SAP MII in Section 3.4.3. Searching and monitoring messages via the message monitor is discussed in Section 3.4.4, and how to clean up the buffered messages from SAP MII is described in Section 3.4.5.

3.4.1 Process Overview

You can synchronize enterprise system data with the manufacturing plant floor systems in two ways:

- In a data pull**, t**he manufacturing integration systems pull the data from the enterprise systems on an as-needed basis.
- In a data push**, t**he data is pushed by the enterprise when required, automatically based on certain events or user actions.

The pull-based method mostly executes RFCs or enterprise services provided by enterprise systems from SAP MII using the SAP Java Connector, SAP Java Resource Adapter, and web service action blocks. The push-based method is often considered necessary because it enables the enterprise system planning and sending data to the plant floor when required. In addition, changes in master data, inventory data, or manufacturing order release can trigger the synchronization of data. The message services module enables users to receive data sent by the enterprise systems. Message services provides the functionality to synchronize enterprise data, such as data from SAP ERP (or SAP S/4HANA) to SAP MII, by asynchronous data transfer. You can use the message services to receive IDoc and RFC messages from SAP systems and also XML messages by HTTP POST from any enterprise systems. The list of inbuilt SAP MII message listeners is provided in Table 3.21.

Message Listeners	Purpose
XMIIIDOC01 to XMIIIDOC10	IDoc listener used for IDoc-based communication
XMIIMESSAGELISTENER	Used for HTTP XML communication
XMIIRFC01 to XMIIRFC10	RFC listener used for RFC asynchronous communication
XMIISRFC01 to XMIISRFC10	Synchronous RFC listener used for synchronous RFC communication

Table 3.21 SAP MII Message Listeners

The messages received in SAP MII are buffered into the system database, and you can either process them automatically according to predefined rules or categorize them for ad hoc processing. The architecture of message services is shown in Figure 3.19.

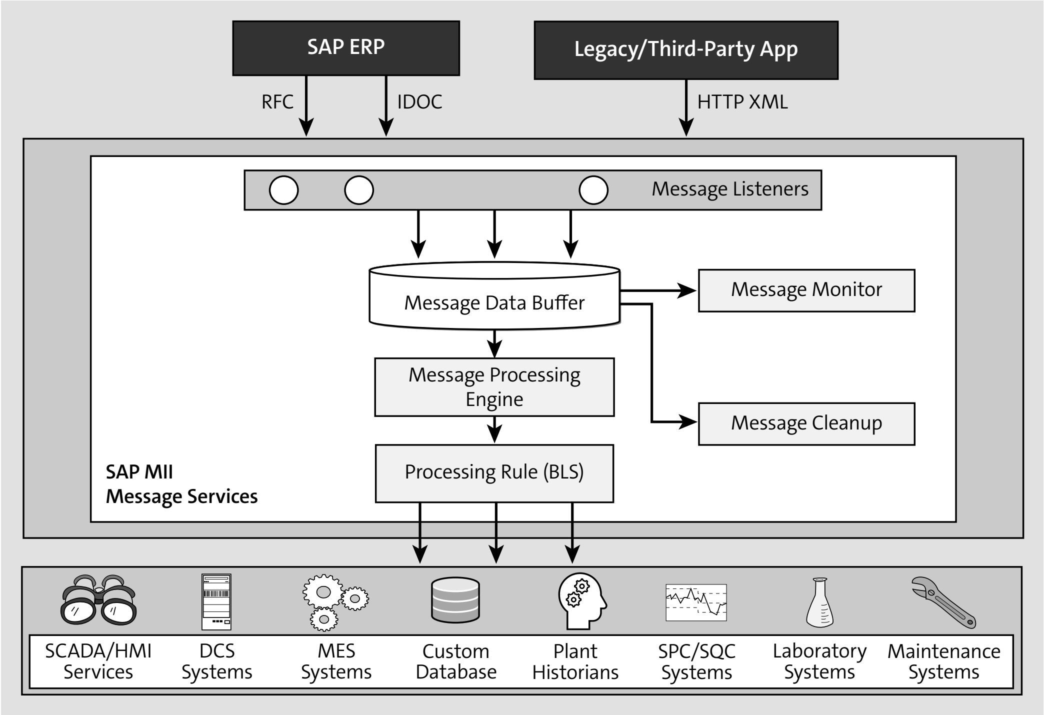


Figure 3.19 SAP MII Message Services

The main functionalities provided by message services in SAP MII are as follows:

- Receive IDoc, RFC, and XML messages by HTTP POST
- Process messages according to processing rule or category
- Monitor buffered messages
- Clean up buffered messages

You can configure message listeners in SAP MII to receive RFC or IDoc messages from enterprise systems asynchronously. The message listeners use SAP Java Connector, and you need to configure different instances of it for each sender system. The message listener for XML messages by HTTP POST is predefined and preconfigured in SAP MII, and you can use it to receive messages from any system which can send XML messages using HTTP POST.

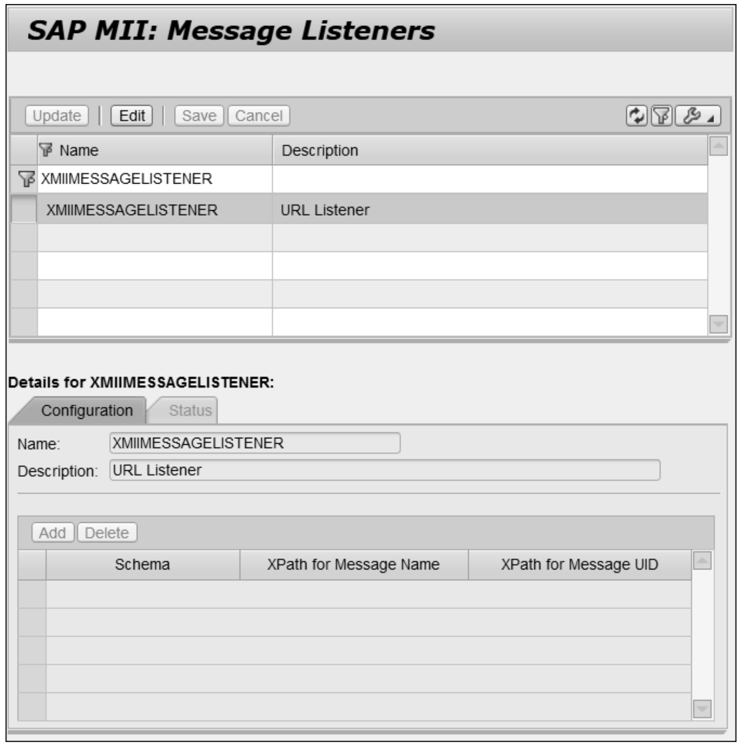
After configuring the message listeners, you can use the processing rule editor to define rules for messages received from different sender systems. A processing rule can either process the message data automatically or assign the message to a category for ad hoc processing. Whenever a message is received by the message listener in SAP MII, it's first saved in the database for buffering and then, if a predefined processing rule exists for it, processed automatically. The messages received and buffered in SAP MII can also be deleted automatically by defining message cleanup rules. The message cleanup rules run periodically for conditionally deleting messages from the database. You can monitor the messages received by the different message listeners from the message monitor and reprocess or delete them individually if required.

Example

You can use message services in many scenarios in SAP MII composite applications. One such example is receiving the LOIPRO IDoc for production orders or control recipe data (by RFC) from SAP ERP/SAP S/4HANA as we discussed in Chapter 1. SAP MII buffers the data as an XML message in the database, and you can use a predefined rule or an ad hoc processing rule to parse the message data, extract the relevant information, and populate it in a custom database table. You can then query this table and populate the production operator dashboard developed in SAP MII, from which the user can view the schedule for a work center and subsequently perform order confirmation.

3.4.2 Configuring the Message Listeners

The message listeners in SAP MII are of four types: two listeners for RFC, XMIIRFC[number] adapter for asynchronous and XMIISRFC[number] adapter for synchronous connection; IDoc messages, which you can configure for each sender system; and the XMIIMESSAGELISTENER message listener, which is a single preconfigured listener for receiving XML messages sent by HTTP POST from any system. Initially, you'll see only a single preconfiguration, for the message listener named XMIIMESSAGELISTENER, as shown in Figure 3.20.



Name	Description
XMIIMESSAGELISTENER	URL Listener

Details for XMIIMESSAGELISTENER:

Name: XMIIMESSAGELISTENER

Description: URL Listener

Schema	XPath for Message Name	XPath for Message UID

Figure 3.20 HTTP XML Message Listener

This is the preconfigured message listener for receiving XML messages by HTTP POST. You can also edit the existing XMIIMESSAGELISTENER and add a schema, XPath for the message name, and XPath for the message UID. After adding the schema, the system will help you get the message name and message UID from there. To send an HTTP XML message to SAP MII, the message needs to be sent to the following URL using HTTP POST as the payload XML:

http://<host>:<port>/XMII/Illuminator?service=WSMessageListener&mode=WSMessageListenerServer&NAME=<UniqueMessageName>&IllumLoginName=<username>&IllumLoginPassword=<password>&Content-Type=text/xml

Replace the *<host>* and *<port>* elements in the URL with the hostname and HTTP port, respectively, of the SAP MII server. Replace the *<UniqueMessageName>* element with the name of the message as specified in the processing rule for the message. If you don't specify a value, the root element name of the XML message is considered the message name, and the processing rule is determined accordingly.

SAP MII also provides quality of services (QoS) functionality where you can specify that messages will be processed exactly once (EO) or exactly once in order (EOIO). The MessageUID is required to process the messages. If you specify EOIO, then the messages will be processed one after another in the queue as those are received. If you specify EO, then messages will be processed in any order inside the queue.

This is the URL for EO:

http://<server>:<port>/XMII/Illuminator?service=WSMessageListener&mode=WSMessageListenerServer&NAME=<UniqueMessageName>&MESSAGEUID=<UniqueMessageUID>&Content-Type=text/xml

This is the URL with sequence and message numbers for EOIO:

http://<server>:<port>/XMII/Illuminator?service=WSMessageListener&mode=WSMessageListenerServer&NAME=<UniqueMessageName>&MESSAGEUID=<UniqueMessageUID>&SEQUENCENAME=<SequenceName>&MESSAGENUMBER=<MessageNumber>&Content-Type=text/xml

You can configure a maximum of 10 RFC and IDoc message listeners (each) as required, which are provided in SAP NetWeaver Administrator. To configure an SAP MII message listener for IDocs or tRFCs, you need to create an RFC destination. Login to the SAP ERP or SAP S/4HANA system and go to Transaction SM59. Now, you can select **TCP/IP Connections** and click the **Create** button. You need to enter a value for **Destination** (the name of the RFC destination), select **T** as the **Connection Type**, and provide a value for **Description**. Next, press the Enter key and move to the **Technical Settings** tab page on the **RFC Destination** screen. Enter the required data as shown in Table 3.22. After entering the required data in the **Technical Settings** tab, you can save the entries.

Field	Description	User Action and Value
Activation Type	Radio buttons	Choose **Registered Server Program**
Registered Server Program: Program ID	Identification of a registered RFC server program	Enter the same name as the RFC destination for the program ID
Gateway Options: Gateway Host	Gateway host name	Enter the SAP system application server
Gateway Service	Gateway service	Enter "sapgw <SAP system number>"

Table 3.22 Creation of RFC Destination

Now, open *http>>://<Host>:<Port>/nwa* and go to the path **Configuration • Infrastructure • Application Resources** to configure the SAP MII message listener. On the **Application**

Resource page select **Resource Adapters** from the **Show** dropdown field. The system will display all available resource adapters in the table. Either you can filter for a specific resource adapter (e.g., `XMIIIDOC01`) by providing the resource adapter name in the **Resource Name** column or you can select the specific resource adapter from table. Now you can go to the **Properties** tab at bottom of the **Resource Details** area. You need to enter the required values as shown in Table 3.23. After you're finished, click the **Save** button.

Field Name	Description
Program ID	SAP S/4HANA program ID, which is defined in Transaction SM59
MaxReaderThreadCount	Maximum count of listening servers
SAPClient	SAP S/4HANA client
UserName	SAP S/4HANA service user name
Password	Password of SAP S/4HANA service user
Language	Language
ServerName	SAP S/4HANA application server
PortNumber	SAP S/4HANA system number

Table 3.23 Resource Adapter Configuration

To test the connection, you need to log onto the SAP S/4HANA system and go to Transaction SM59. Now expand **TCP/IP Connections** and select the RFC destination you have created earlier. Next, click the **Display** icon. The system will display the RFC destination, and you can click **the Connection Test** button. A successful connection will be displayed for the connection test event. Apart from the RFC destination configuration and connection test, you also need to perform required ALE configuration on the SAP S/4HANA side to perform the IDoc communication from SAP S/4HANA to SAP MII.

> **Note:**
>
> After performing the resource adapter configuration and successful connection testing from the RFC destination on the SAP S/4HANA side, you need to log into SAP MII using URL *http>>://<Host>:<Port>/XMII/Menu.jsp* and the go to menu path **Message Services • Message Listeners.** On the **Message Listeners** page, you need to select configured message listeners (e.g., `XMIIIDOC01`, which is configured in the **Resource Adapter** settings) and click the **Update** button to update the configuration of the message listener. On the **Status** tab page, you'll see the connection status.

3.4.3 Configuring the Processing Rules for Messages

Processing rules in message services are defined to process messages automatically when received from an external sender system. The processing rules can be existing BLS transactions that you can use to process the message data received or to send the message data down to manufacturing plant floor systems. A processing rule is defined for each combination of a message listener and a message name. You can define two types of processing for the messages: transaction-based and category-based.

In transaction-based processing, you need to specify a BLS transaction as the processing rule. In category-based processing, you need only to assign the message to a category (no automatic processing is done). This means that when a specific message is received by the message listener, it's buffered into the database, and the corresponding processing rule is searched. If a BLS transaction is found as the processing rule, then that transaction is executed, passing the XML message data as an input parameter as specified in the processing rule. If a category is found for the processing rule, then the message is buffered into the database and assigned to a category. You can use the message category to logically group messages for ad hoc processing. You can use the message data and other details (e.g., received timestamp, message listener, message name, etc.) in the message services actions in BLS, where the messages can be selected for a specific category as well.

For example, you can use transaction-based processing when you receive production order details from SAP ERP or SAP S/4HANA. SAP MII sends them to the MES for execution, using a BLS as a processing rule. You can use category-based processing where SAP MII doesn't send the messages to any system and buffers the message in the message service data store with a category or tag., You can provide a BLS transaction where the messages are queried based on certain criteria using the message service action blocks (explained in Chapter 6, Section 6.4.8), which an external system can execute as a web service to read the messages received and buffered in SAP MII.

To configure the processing rules, open the **Message Processing Rules** page from the **Message Services** menu category in SAP MII. If no processing rule has been created before, then the message processing rule table doesn't contain any items. Now let's discuss how to configure transaction-based and category-based processing rules. On the **Message Processing Rules** page, click the **Create** button. The system will open a detail page where you can configure the processing rules as described in Table 3.24.

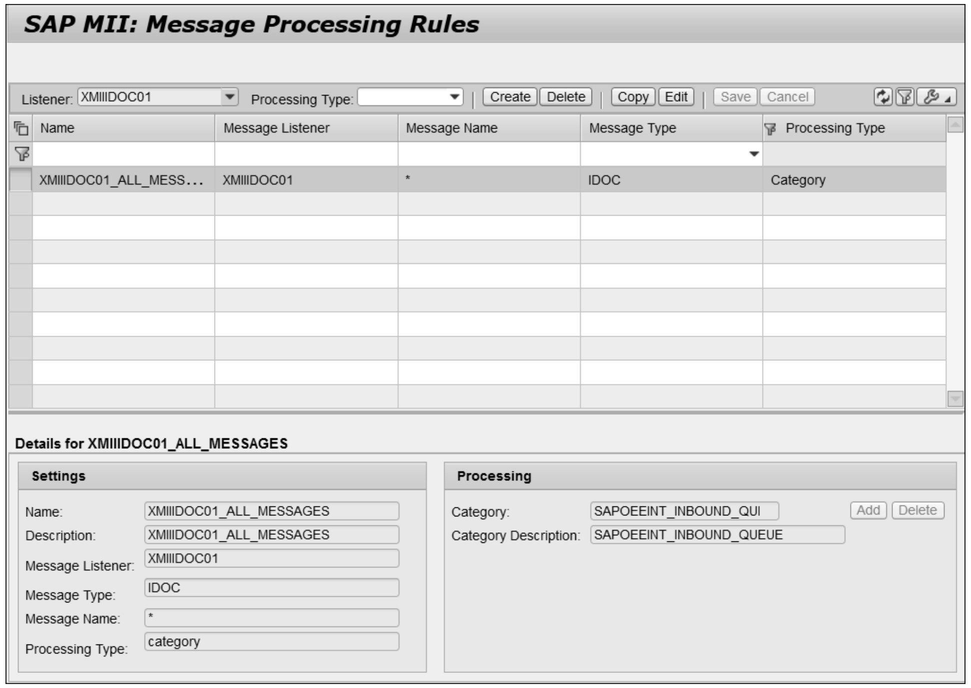
Field Name	Required Configuration
Name	Unique name of processing rule.
Description	Description of processing rule.

Table 3.24 Message Processing Rules

Field Name	Required Configuration
Message Listener	Select the message listener server from available list to which the rule will be applied.
Message Type	The message type will be selected automatically based on message listener selection. For example, message type **IDOC** is selected for message listener **XMIIIDOC01** to **XMIIIDOC10**; message type **Web Service** is selected for message listener **XMIIMESSAGELISTENERS**; and message type **RFC** is selected for **XMIIRFC01** to **XMIIRFC10** and **XMIISRFC01** to **XMIISRFC10**.
Message Name	Specify the message names for IDoc, web service, and RFC messages. You can also provide the value * for the message name, which applies to all messages.
Processing Type	Here, you can select the transaction or category processing type. In the *transaction* processing type, messages are processed immediately through a transaction. In the *category* processing type, messages are persisted in a persistent queue where you can use the message services actions from the SAP MII workbench to retrieve those messages. If you select the **SRFC** message type, then you won't have the option to select a category processing type.
Processing Section	The fields in processing sections are dynamically available based on processing type selection from the **Processing Type** field.
Transaction	This field will be available if you select a transaction processing type. You can browse the transaction created in the SAP MII workbench.
Persist Transaction	You can select an appropriate option here: **Always**, **On Error**, or **Never**.
Log Level	Here you can select **Unknown**, **None**, **Debug**, **Info**, **Warn**, **Error**, or **Fatal**.
Parameters	The input properties are available as parameters based on selected message listener. These parameters can be mapped to the message fields such as **ReceivedMessageXML**, **MessageUID**, and **MessageID**.
Category	If you select **Category** for **Processing Type**, then this field is available in the **Processing** section. It groups messages and saves them in the buffer for sequential processing. Assigning messages to a category works like a virtual queue.
Description	Holds the description of the category.

Table 3.24 Message Processing Rules (Cont.)

As per the description and your business requirements, you can configure the message processing rules as shown in Figure 3.21.



SAP MII: Message Processing Rules

Listener: XMIIIDOC01

Processing Type:

Name	Message Listener	Message Name	Message Type	Processing Type
XMIIIDOC01_ALL_MESS...	XMIIIDOC01	*	IDOC	Category

Details for XMIIIDOC01_ALL_MESSAGES

Settings

Name: XMIIIDOC01_ALL_MESSAGES
Description: XMIIIDOC01_ALL_MESSAGES
Message Listener: XMIIIDOC01
Message Type: IDOC
Message Name: *
Processing Type: category

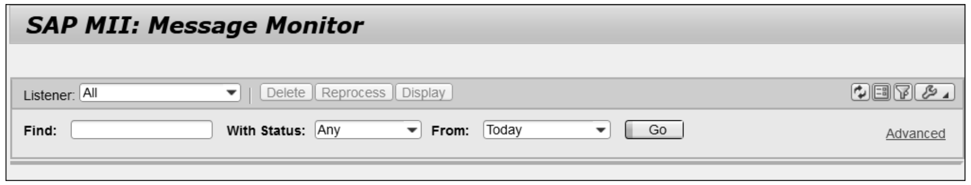
Processing

Category: SAPOEEINT_INBOUND_QUI
Category Description: SAPOEEINT_INBOUND_QUEUE

Figure 3.21 Message Processing Rules

3.4.4 Using the Message Monitor

You can use the message monitor provided in message services to monitor the status of the messages received. You can also perform additional functions, such as reprocessing, displaying, and deleting messages. The message monitor menu provides the monitoring options for all types of messages. The default selection options in the message monitor display all messages received by any message listener for the last 24 hours (today). You can search messages based on **Listener** selection from the available list, by entering a value (the name of the message you want to find) in the **Find** input field, by selecting an option in the **With Status** dropdown, and by using the **From** field value selection. After specifying search criteria in the required field, you can click the **Go** button to perform your search, as shown in Figure 3.22.



SAP MII: Message Monitor

Listener: All

Find:

With Status: Any

From: Today

Figure 3.22 Message Monitor Search

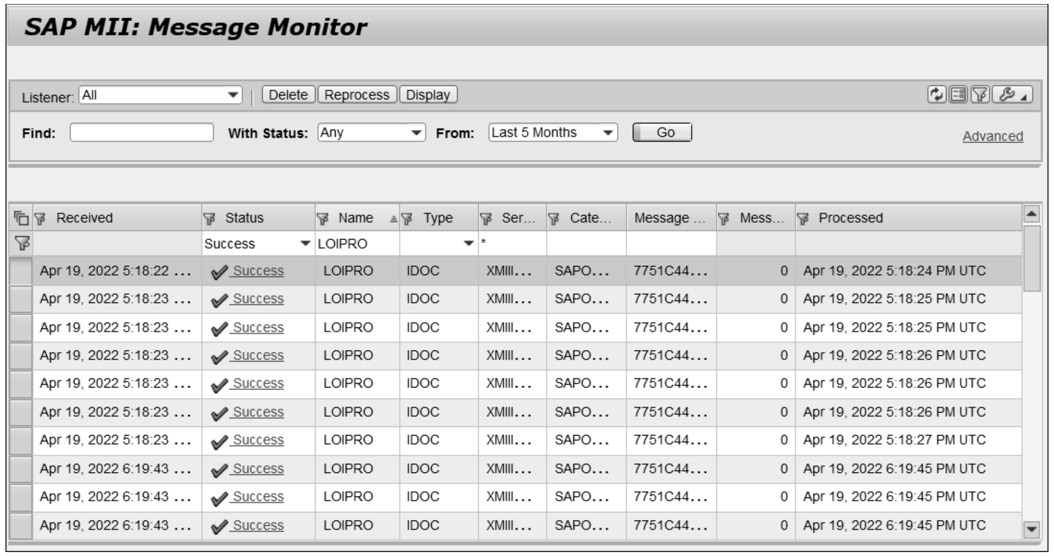
In the advanced search criteria, you can specify the details listed in Table 3.25.

Field Name	Usage
Listener	Select the message listener you want to search; select **All** to search all message listeners
Name	Specify the name of the message you want to search
Category	Select the category of messages you want to search
From	Select a start date for your search criteria
To	Select an end date for your search criteria
Status	Specify the message status you want to search for using multiple checkbox selection, such as **Received**, **Success**, **Failed**, **No Rule**, **Categorized**, **Running**, or **Duplicate**
Type	Select the message type you want to search for, such as **RFC**, **IDOC**, or **Web Service**

Table 3.25 Advance Search of Message Monitor

After searching, the available messages will be displayed in the message table with columns such as **Received**, **Status**, **Name**, **Type**, **Server**, **Category**, **Message UID**, **Message Number**, and **Processed**, as shown in Figure 3.23. To filter the message display, you can use the following selection criteria:

- **Status**
  Select the specific message status from the dropdown to filter messages on that status. Available options are **Received, Success, Failed, No Ruled, Categorized, Running** and **Duplicate**.
- **Name**
  Select specific message names (e.g., LOIPRO) to display only those types of messages.
- **Type**
  Select specific message types such as IDoc, RFC, and web service to display only those types of messages.
- **Server**
  Select the specific message listener from the dropdown. Selecting an asterisk signifies all message listeners defined in SAP MII.
- **Category**
  Select message categories created in the processing rules to display only messages assigned to that category. An asterisk signifies all categories or no categories.
- **Message UID**
  You can filter messages based on the unique identification number of the message.



SAP MII: Message Monitor

Listener: All

Find:
With Status: Any
From: Last 5 Months

Received	Status	Name	Type	Ser…	Cate…	Message …	Mess…	Processed
	Success	LOIPRO		*				
Apr 19, 2022 5:18:22 …	✓ Success	LOIPRO	IDOC	XMIII…	SAPO…	7751C44…	0	Apr 19, 2022 5:18:24 PM UTC
Apr 19, 2022 5:18:23 …	✓ Success	LOIPRO	IDOC	XMIII…	SAPO…	7751C44…	0	Apr 19, 2022 5:18:25 PM UTC
Apr 19, 2022 5:18:23 …	✓ Success	LOIPRO	IDOC	XMIII…	SAPO…	7751C44…	0	Apr 19, 2022 5:18:25 PM UTC
Apr 19, 2022 5:18:23 …	✓ Success	LOIPRO	IDOC	XMIII…	SAPO…	7751C44…	0	Apr 19, 2022 5:18:26 PM UTC
Apr 19, 2022 5:18:23 …	✓ Success	LOIPRO	IDOC	XMIII…	SAPO…	7751C44…	0	Apr 19, 2022 5:18:26 PM UTC
Apr 19, 2022 5:18:23 …	✓ Success	LOIPRO	IDOC	XMIII…	SAPO…	7751C44…	0	Apr 19, 2022 5:18:26 PM UTC
Apr 19, 2022 5:18:23 …	✓ Success	LOIPRO	IDOC	XMIII…	SAPO…	7751C44…	0	Apr 19, 2022 5:18:27 PM UTC
Apr 19, 2022 6:19:43 …	✓ Success	LOIPRO	IDOC	XMIII…	SAPO…	7751C44…	0	Apr 19, 2022 6:19:45 PM UTC
Apr 19, 2022 6:19:43 …	✓ Success	LOIPRO	IDOC	XMIII…	SAPO…	7751C44…	0	Apr 19, 2022 6:19:45 PM UTC
Apr 19, 2022 6:19:43 …	✓ Success	LOIPRO	IDOC	XMIII…	SAPO…	7751C44…	0	Apr 19, 2022 6:19:45 PM UTC

Figure 3.23 Monitor Messages Display

To display a selected message from the table in a new window, click the **Display** button from the **Message** Monitor page. As per business needs, you can also reprocess a message and delete a message from message table using the **Reprocess** and **Delete** buttons.

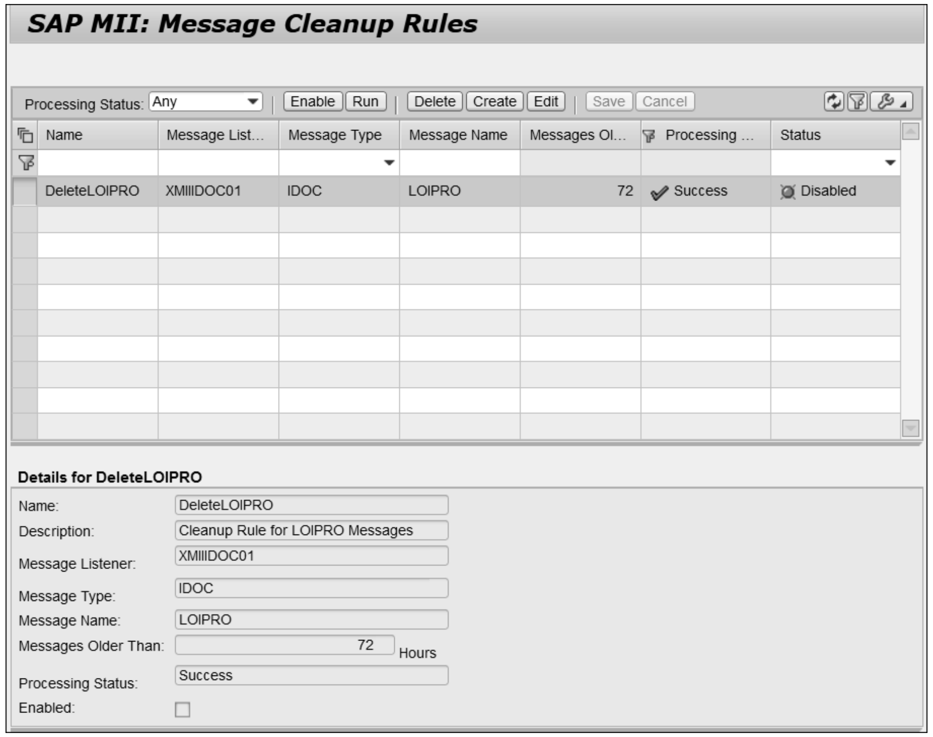
> **Warning**
>
> Deleting messages permanently deletes them from the SAP MII database. They cannot be retrieved again.

In this section, you learned about how to search and monitor messages in SAP MII. Next let's discuss configuration of message cleanup rules.

3.4.5 Configuring and Using the Message Cleanup Rules

Received messages are buffered in the SAP MII database by default, regardless of whether or not any processing rule is defined for them. Messages buffered in SAP MII are not deleted automatically, but you can define message cleanup rules. The message cleanup rules can be based on the message listener, message type, message name, message age, and processing status. The message cleanup rules run periodically at a specific time interval, which you can specify in **System Management • System Properties • Message Cleanup Interval[Hours]**.

To define a new message cleanup rule, open the editor for message cleanup rules (Figure 3.24) under the **Message Services** category. You can specify a unique message cleanup rule **Name** and an optional **Description**. Also select the specific message listener from the **Server Name** dropdown, or select an asterisk for any message listener. Select the **Message Type** as **IDoc**, **RFC**, or **Web Service**, and specify the **Message Name** as either a specific message name or with an asterisk for any message. In the **Message Older Than** property, specify the time in hours. Messages buffered in SAP MII longer than this specified time are deleted by the cleanup rule if other criteria are satisfied. You can specify the **Processing Status** property as either **All**, which signifies that messages having any of the statuses in the list will be deleted, or any of the usual message processing statuses that are available, except the **Running** status.



Processing Status: Any

Name	Message List...	Message Type	Message Name	Messages Ol...	Processing ...	Status
DeleteLOIPRO	XMIIIDOC01	IDOC	LOIPRO	72	✓ Success	Disabled

Details for DeleteLOIPRO

Name: DeleteLOIPRO
Description: Cleanup Rule for LOIPRO Messages
Message Listener: XMIIIDOC01
Message Type: IDOC
Message Name: LOIPRO
Messages Older Than: 72 Hours
Processing Status: Success
Enabled: □

Figure 3.24 Message Cleanup Rule

You can view one or multiple message cleanup rules in the **Message Cleanup Rules** table, where you can enable or disable and execute the cleanup rules. Using the **Edit** button, you can edit the cleanup rule properties, such as **Rule Description**, **Server Name**, **Message Name**, **Age**, **Processing Status**, and **Rule Activation Status**. You can enable or disable one or more rules by clicking the **Enabled** or **Disabled** buttons. In addition, you can execute the selected rules on an ad hoc basis by clicking the **Run Rule** button. You can configure the message cleanup run interval (**Message Cleanup Interval**) from SAP MII system properties. You can enter a value (in hours) for how often the message cleanup rule will run.

The messages received by the message listeners can also be retrieved and processed by BLS using the message services action blocks, which are explained further in Chapter 6.

3.5 Summary

In this chapter, you learned how to configure the data servers and message services that help you to connect SAP MII with plant floor systems and receive messages from enterprise systems.

In the next chapter, you'll learn about how to develop composite applications on SAP MII, work with the SAP MII workbench, and create query templates using the configured data servers.

Contents

3 Managing Data from External Systems 83

4 Developing Composite Applications 145

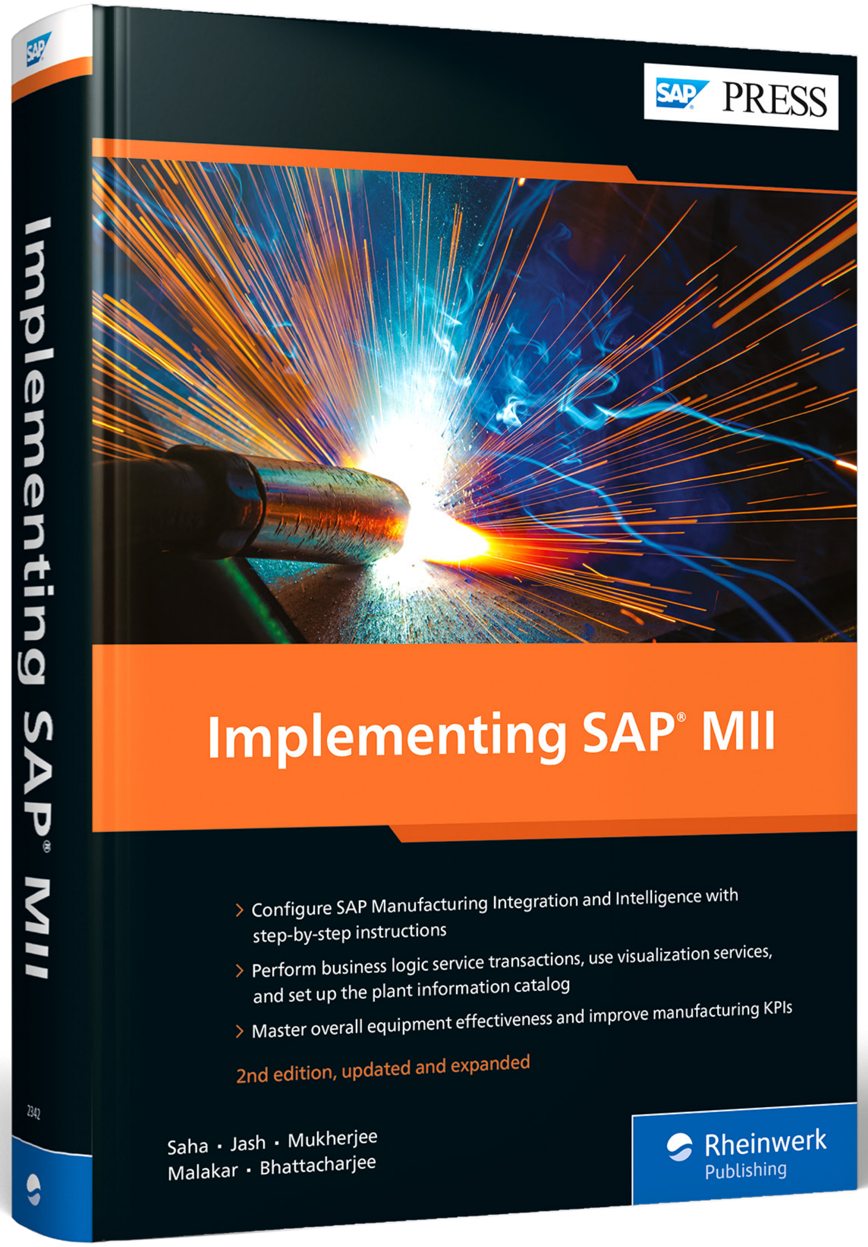
5 Manufacturing Data Objects 183

6 Business Logic Service Transactions 201

7 Visualization Services 329

8 Plant Information Catalog 413

Index


SAP PRESS
Implementing SAP® MII
› Configure SAP Manufacturing Integration and Intelligence with step-by-step instructions
› Perform business logic service transactions, use visualization services, and set up the plant information catalog
› Master overall equipment effectiveness and improve manufacturing KPIs
2nd edition, updated and expanded
Saha · Jash · Mukherjee
Malakar · Bhattacharjee
Rheinwerk Publishing
Implementing SAP® MII

Saha, Jash, Mukherjee, Malakar, Bhattacharjee

Implementing SAP MII

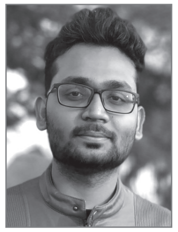
705 pages | 01/2023 | $99.95 | ISBN 978-1-4932-2342-8

www.sap-press.com/5610

Dipankar Saha is an architect for SAP manufacturing and Industry 4.0 solutions and has worked in several SAP MII implementation projects and also in the product design and development of SAP MII. Dipankar is an SAP-certified associate enterprise architect. He has more than 20 years of experience in the IT industry and has a bachelor's degree in chemical engineering.

Chandan Jash is a technical architect and SME for SAP ME and SAP MII implementation projects at IBM India. He has worked on many SAP manufacturing implementations, supported SAP MII and SAP ME projects globally in different roles, and has implemented Industry 4.0 standards. He has worked with SAP solutions for more than 17 years and has a bachelor's degree in computer science and engineering.

Sudipta Mukherjee is an application architect at IBM India with more than 17 years of IT experience. His main areas of expertise are SAP MII, SAP OEE Management, SAP ME, SAP Digital Manufacturing Cloud, SAP BTP, and SAP Internet of Things. He works with global clients to define SAP manufacturing implementation architectures and manage end-to-end delivery. He has a bachelor's degree in computer science and engineering.



Supriya Malakar is a senior consultant at IBM India. He works with global clients for large-scale SAP MII and SAP ME solution implementation. He also has deep expertise in SAP Business Technology Platform and SAP Digital Manufacturing Cloud. He has more than six years of experience in the IT industry and has a bachelor's degree in computer science and engineering.

Abesh Bhattacharjee has been working as an IT consultant, a go-to market leader, and an architect for more than 20 years. He is currently the senior solution architect at Neptune Software responsible for North America.