CHAPTER 3

Basics of SAP MII

Before diving deep into the SAP MII technically and correlating it with the functional requirements, the basic conceptual integral configurations and intelligent development IDE and its functions should be clear to the professionals. This chapter explains the basics of SAP MII, the steps related to its various configurations, and its development components and other basic features.

To work with the technical part of the SAP MII, the important features to know include these:

- Data Services, which are responsible for configuring all the connections to other systems.
- Message Services, which handle message flow in and out through SAP MII.
- The development components, which include logic services, query templates, MDO, visualizations, and other web development components.

Data Services

Data Services provide manufacturing shopfloor connectivity by providing connectors for vendor-specific protocols for plant systems and standard protocols to connect to the relational database. Data connectors come with SAP MII installation that's able to connect to any manufacturing shopfloor system. It also provides different types of configurable data queries (can be configured through SAP MII Workbench) to retrieve or write data to data sources. These data queries can be accessed by HTTP calls or by Java applets from web-based UIs.

Available menu options for Data Services are as follows:

- Data Servers
- Connection Status
- Connections
- Simulator Services

S. Mukherjee and S. Mukherjee (Das), *SAP MII*, DOI 10.1007/978-1-4842-2814-2_3

- Shared Memory
- Time Periods
- Time Period Schedules
- Data Buffer

Some of these features are described in the following sections.

Data Servers

Data servers are used to configure query templates to query data from different sources. You can browse this data can be in the SAP MII Workbench. See Figure 3-1.

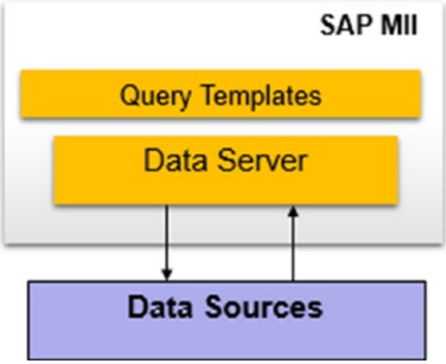


Figure 3-1. *Data server placement in SAP MII*

Server parameters that are common to configure include:

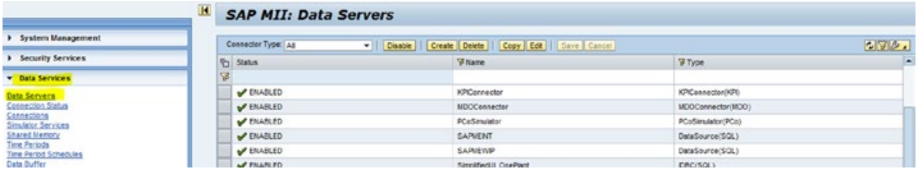
- *Name*: Name of the server provided. It has to be unique.
- *Connector*: To configure the data servers, certain connectors are available: Aggregate, AlarmSuite, DataSource, IDBC, IP21, and IP21OLEDB, InSQL, OLAP, OLEDB Open, Simulator, Universal Data Connector (UDC), Virtual Servers, VirtualAlarm, VirtualIDBC, VirtualOLAP, VirtualTAG, VirtualXML, VirtualXacute, XML, and Xacute.
- *Connector Type*: Type of connection determined by the data source class. The possible connector types are AGG (Aggregate), ALARM, MDO, OLP (OLAP), PCo, SQL, TAG, XCT, XML, and KPIConnector.

- *Enabled*: Indicator used to activate the server. You can check or uncheck it to enable or disable the activation of the server, respectively. If a data server is enabled and you make changes to its parameters, the server is reloaded, and the updated settings are effective immediately.
- *Allow Dynamic Query*: Indicator used to support dynamic queries for users with the Dynamic Query role. When deselected, the data server does not support dynamic queries.

Creating a Connector

Follow these general configuration steps to create a connector:

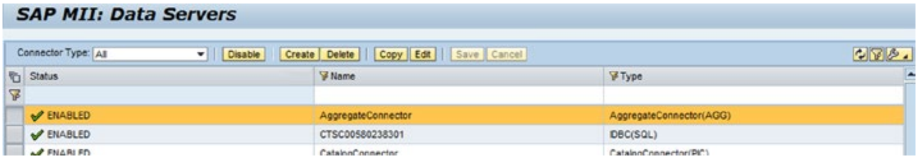
1. From the left menu, select Data Services ➤ Data Servers, as shown in Figure 3-2.



Status	Name	Type
✓ ENABLED	KPIConnector	KPIConnector(KPI)
✓ ENABLED	MDOConnector	MDOConnector(MDO)
✓ ENABLED	PCoSimulator	PCoSimulator(PCo)
✓ ENABLED	SAPMENT	DataSource(SQL)
✓ ENABLED	SAPMEWIP	DataSource(SQL)
✓ ENABLED	SimplifiedUI_OnePlant	IDBC(SQL)

Figure 3-2. *SAP MII Data Server configuration*

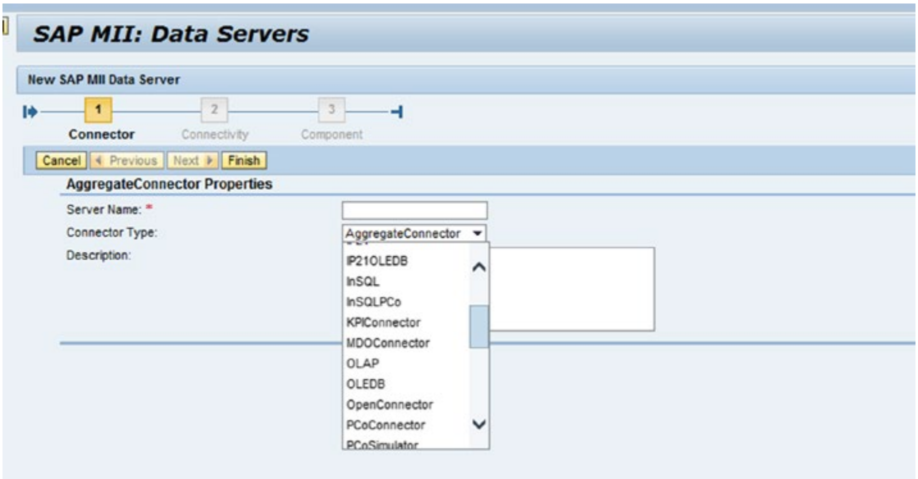
2. Select Create in the next window, as shown in Figure 3-3.



Status	Name	Type
✓ ENABLED	AggregateConnector	AggregateConnector(AGG)
✓ ENABLED	CTSC00580238301	IDBC(SQL)
✓ ENABLED	CatalogConnector	CatalogConnector(PIC)

Figure 3-3. *The Create button in the Data Servers configuration*

3. Give it a proper server name (mandatory) to recognize later on and a description (optional). Select the appropriate connector type from the drop-down, as shown in Figure 3-4.



AggregateConnector Properties

Server Name: *

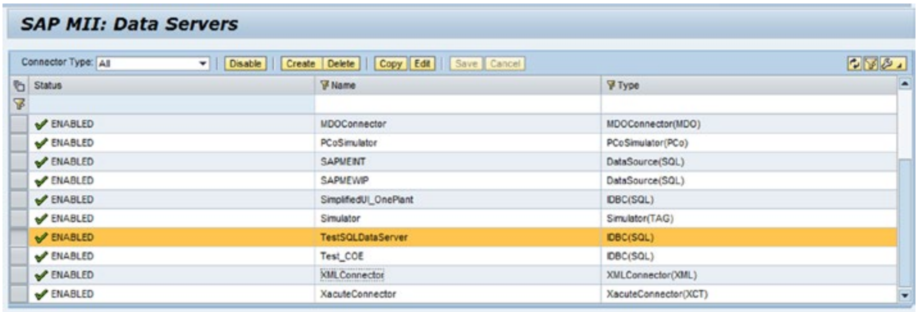
Connector Type: AggregateConnector

Description:

Figure 3-4. *Data Server configuration*

4. Follow the steps for the type of data server you want to create as per your requirements.

5. You can also delete, copy, and edit the selected connector by using the buttons shown in Figure 3-5.

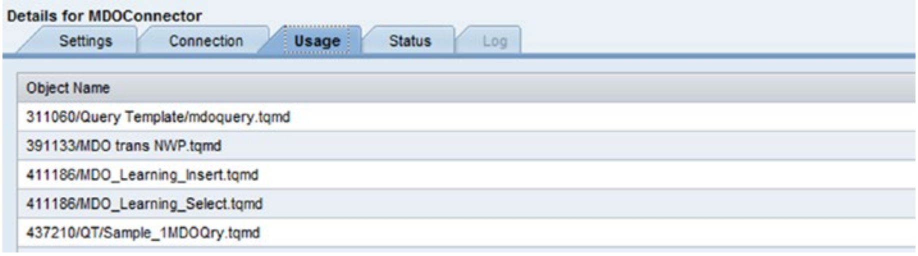


Connector Type: All

Status	Name	Type
✓ ENABLED	MDOConnector	MDOConnector(MDO)
✓ ENABLED	PCoSimulator	PCoSimulator(PCo)
✓ ENABLED	SAPMEINT	DataSource(SQL)
✓ ENABLED	SAPMEWIP	DataSource(SQL)
✓ ENABLED	SimplifiedUI_OnePlant	IDBC(SQL)
✓ ENABLED	Simulator	Simulator(TAG)
✓ ENABLED	TestSQLDataServer	IDBC(SQL)
✓ ENABLED	Test_COE	IDBC(SQL)
✓ ENABLED	XMLConnector	XMLConnector(XML)
✓ ENABLED	XacuteConnector	XacuteConnector(XCT)

Figure 3-5. *Other buttons in SAP MII Data Servers configuration*

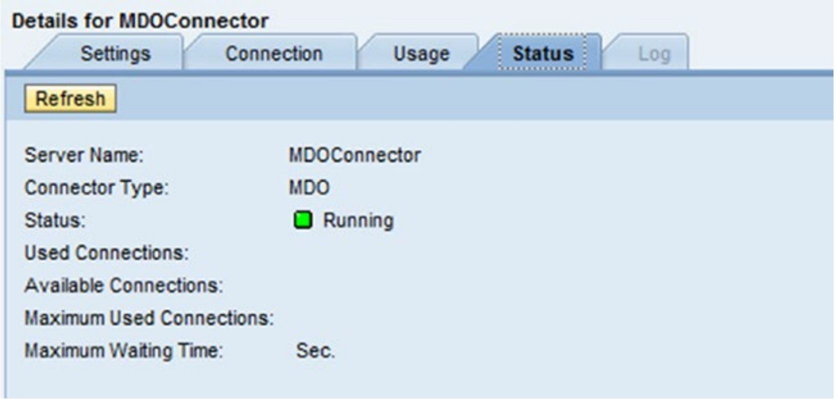
6. Once the data server is selected from the list, details of the connector will be shown and the Usage tab denotes the object names of SAP MII Workbench where the connector is used. See Figure 3-6.



Object Name
311060/Query Template/mdoquery.tqmd
391133/MDO trans NWP.tqmd
411186/MDO_Learning_Insert.tqmd
411186/MDO_Learning_Select.tqmd
437210/QT/Sample_1MDOQry.tqmd

Figure 3-6. *Usage check in the data server*

7. The Status tab shows the status of the connector as Running, Stopped, or Error, as shown in Figure 3-7.



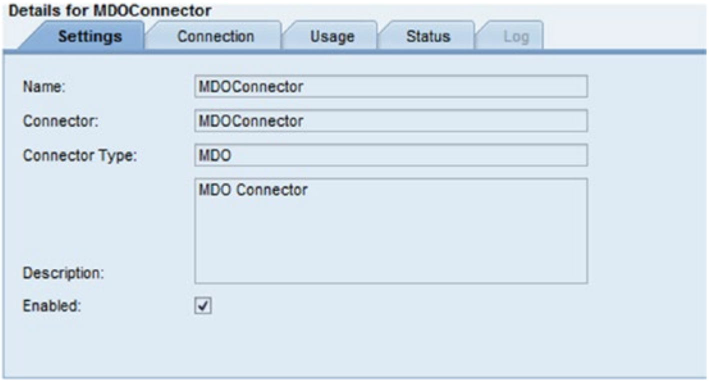
Server Name: MDOConnector
Connector Type: MDO
Status: 🟩 Running
Used Connections:
Available Connections:
Maximum Used Connections:
Maximum Waiting Time: Sec.

Figure 3-7. *Status check in Data Server*

MDO Connector

A connector used to run queries against manufacturing data objects (MDO).

From Step 2 of the general configuration steps, select the connector as MDOConnector. Once that is created, it will be shown in the Settings tab, as shown in Figure 3-8.



Name: MDOConnector

Connector: MDOConnector

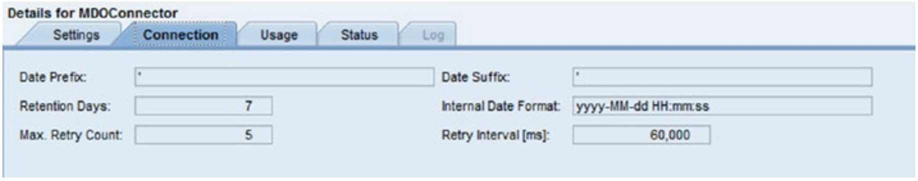
Connector Type: MDO

Description: MDO Connector

Enabled: ☑

Figure 3-8. *Configuring the MDO connector*

The Connection tab shows these settings, as shown in Figure 3-9:



Date Prefix: '

Date Suffix: '

Retention Days: 7

Internal Date Format: yyyy-MM-dd HH:mm:ss

Max. Retry Count: 5

Retry Interval [ms]: 60,000

Figure 3-9. *MDO Connector Connection creation*

- Data Prefix: Defines the prefix used when working with Date or Time database columns.
- Date Suffix: Defines the suffix used when working with Date or Time database columns.
- Retention Days: Used for data buffering. Number of days the execution attempt remains in the data buffer before being removed.
- Max. Retry Count: Used for data buffering. Number of times the system retries the execution attempt before stopping.
- Retry Interval: Used for data buffering. Number of milliseconds the system waits before retrying the execution attempt.

IDBC (SQL/Oracle) Connector

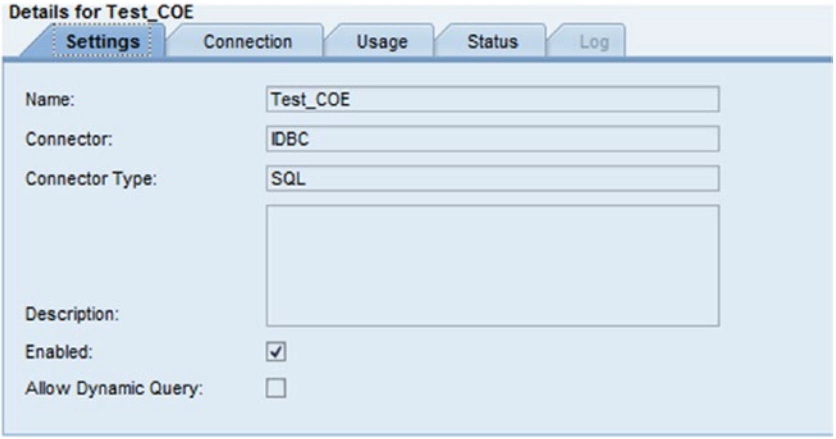
IDBC Connector provides direct connection to any database to create SQL queries on its table data. The examples of databases are Microsoft SQL server, Oracle database, MaxDB, DB2, and so on.

Properties, UserName, Password, and ServerURL for authentication need to be changed at least while creating a new IDBC Data Server configuration.

The JDBC drivers included with SAP MII are the following:

- *Oracle*: You must use Oracle 7.3.4 or newer (oracle.jdbc.driver.OracleDriver).
- *SQL Server*: You must use SQL Server 6.X, 7.X or 2000 (com.inet.tds.TdsDriver).
- ODBC

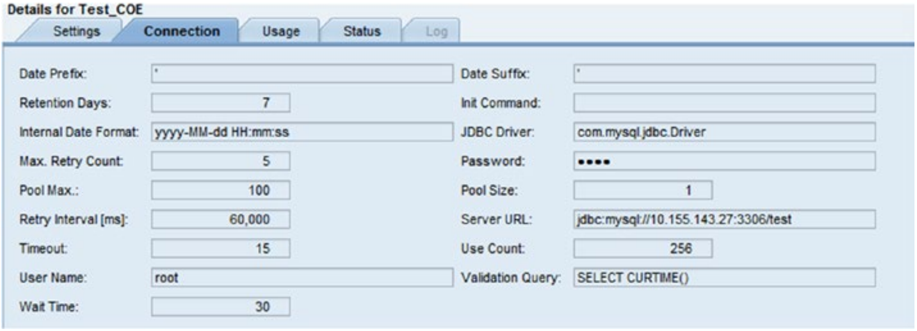
From Step 2 of general configuration, select the connector as MDOConnector. After it's created, it appears in Settings tab as shown in Figure 3-10.



Name:	Test_COE
Connector:	IDBC
Connector Type:	SQL
Description:	
Enabled:	☑
Allow Dynamic Query:	□

Figure 3-10. *The IDBC connector setting*

From the Connection tab, it appears as shown in Figure 3-11.



Date Prefix:	'	Date Suffix:	'
Retention Days:	7	Init Command:	
Internal Date Format:	yyyy-MM-dd HH:mm:ss	JDBC Driver:	com.mysql.jdbc.Driver
Max. Retry Count:	5	Password:	••••
Pool Max.:	100	Pool Size:	1
Retry Interval [ms]:	60,000	Server URL:	jdbc:mysql://10.155.143.27:3306/test
Timeout:	15	Use Count:	256
User Name:	root	Validation Query:	SELECT CURTIME()
Wait Time:	30		

Figure 3-11. *IDBC Connection configuration*

Here are some additional important connection configuration options available for IDBC connectors:

- *Pool Size*: The connection pool is the total number of threads any DB connection creates so that it can allow that many numbers of parallel DB hits. Pool size indicates the number of connections MII can open at a time in parallel. The default value for the pool size is 1.

- *Pool Max*: Users can set the maximum pool size based on which the DB server allows users to create the pool connection. User can use the same value defined for the DB connection pool size. By defining this in MII, it can also open a maximum of this number of connections at a time. The default value is 100.

- *Wait Time*: Defines the maximum wait time for a connection if the pool is full.

For further information on Data Servers, refer to the following SCN blogs written by Suman Mukherjee: `https://blogs.sap.com/2014/02/24/data-connection-details-for-data-server-configuration/` and `https://blogs.sap.com/2013/03/13/importance-of-pool-size-configuration-of-data-server-in-sap-mii/`.

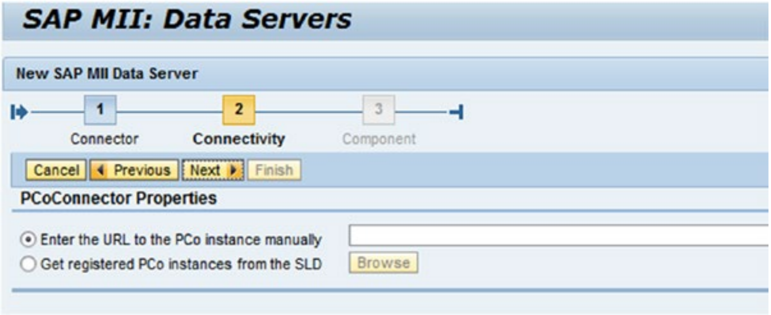
The PCo Connector

The Plant Connectivity (PCo) connector allows SAP Manufacturing Integration and Intelligence (SAP MII) to support the query capabilities of PCo. See Figure 3-12.

When creating a PCo connection, one of the following options can be followed:

- Manually enter the URL to the PCo management service

- Retrieve the registered PCo instances from the System Landscape Directory (SLD)



PCoConnector Properties

● Enter the URL to the PCo instance manually

○ Get registered PCo instances from the SLD

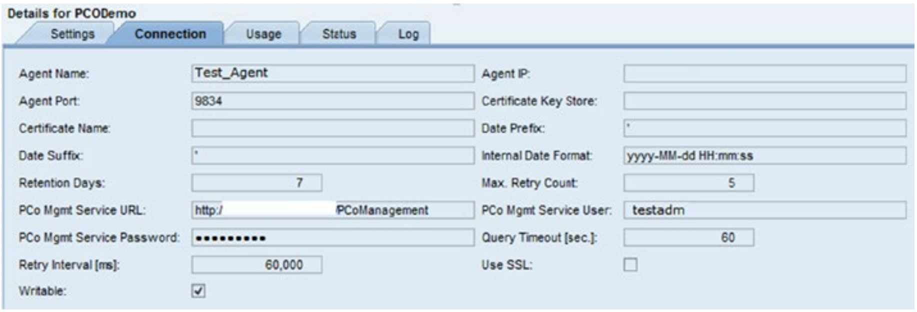
Figure 3-12. *PCo connector configuration*

To configure the PCo connector properties as shown in Figure 3-12, two options are available:

- Option 1: Complete the initial setup of the functional unit SLD using CTC templates. This is a standard procedure of the SAP NetWeaver SLD client configuration. Register the PCo management service in the SLD. After this, follow the instructions below.
- Option 2: If the PCo management Web service interface requires authentication, users are prompted to enter a username and password. After this, follow the instructions below.

To create a PCo data server, you need to go to MII Menu ➤ Data Services ➤ Data Servers. You can choose your connector type as PCoConnector and click on the Create button. You need to provide a unique name for the server.

On the next page, system will ask for the URL to the PCo instance. You need to provide the URL, such as `http://<Host>:<Port>/PCoManagement`. (`Host` is the system name where the PCo is installed and `Port` is the default port `50050` configured in the PCo Management Console).



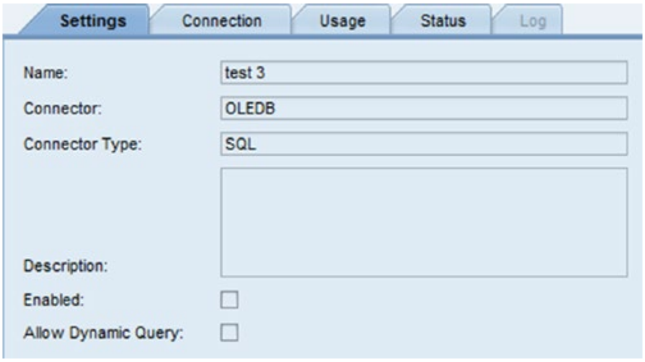
Agent Name:	Test_Agent	Agent IP:	
Agent Port:	9834	Certificate Key Store:	
Certificate Name:		Date Prefix:	'
Date Suffix:	'	Internal Date Format:	yyyy-MM-dd HH:mm:ss
Retention Days:	7	Max. Retry Count:	5
PCo Mgmt Service URL:	http:/[redacted]PCoManagement	PCo Mgmt Service User:	testadm
PCo Mgmt Service Password:	••••••••••	Query Timeout [sec.]:	60
Retry Interval [ms]:	60,000	Use SSL:	□
Writable:	☑		

Figure 3-13. *PCo connection configuration*

At the end, you need to provide a user and password.

UDC (OLEDB) Connector

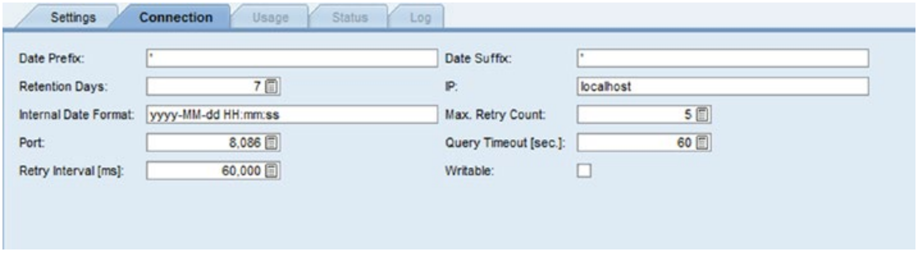
The universal data connector (UDC) is a framework that allows access to SAP Manufacturing Integration and Intelligence (SAP MII) services through proprietary server applications that you develop. The OLEDB Connector is a UDC that allows access to OLE DB data sources. The OLEDB Connector provides OLEDB access to Microsoft-compliant data sources using the PCo server. Tag queries can be created using OLEDB Data Server by selecting the connector as OLEDBConnector. After it is created, it is available in the Settings tab, as shown in Figure 3-14.



Name:	test 3
Connector:	OLEDB
Connector Type:	SQL
Description:	
Enabled:	□
Allow Dynamic Query:	□

Figure 3-14. *The OLEDB Connector Settings tab*

From the Connection tab, you'll see the settings shown in Figure 3-15.



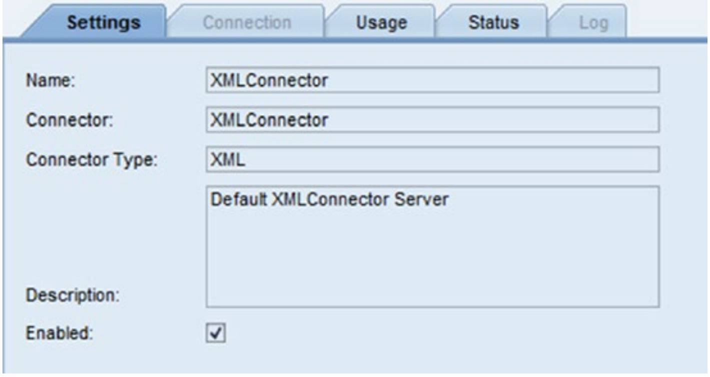
Date Prefix:	'	Date Suffix:	'
Retention Days:	7	IP:	localhost
Internal Date Format:	yyyy-MM-dd HH:mm:ss	Max. Retry Count:	5
Port:	8,086	Query Timeout [sec.]:	60
Retry Interval [ms]:	60,000	Writable:	□

Figure 3-15. *OLEDB Connection configuration*

Additionally, select the Writable checkbox. If the checkbox is selected, the connector will support writing to the OLEDB agent. If the writable checkbox is not selected, you will not be able to write to the agent, which means the `CurrentWrite` and `HistoryWrite` queries will fail.

The XML Connector

The XML Connector can be used to connect to XML-compliant data sources to retrieve the data in SAP MII XML format. A default XML Data Server is available in SAP MII named XMLConnector, although additional instances of XML Data Server can be also be configured. See Figure 3-16.



Name: XMLConnector

Connector: XMLConnector

Connector Type: XML

Description: Default XMLConnector Server

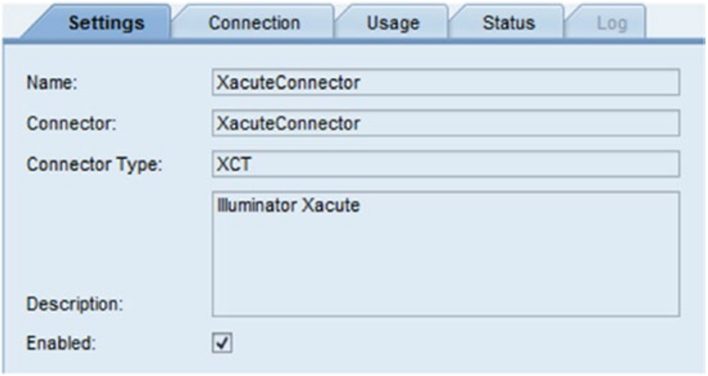
Enabled: ☑

Figure 3-16. *The XML Connector setting*

In the XML Data Server configuration, only the Description property can be edited. Using the XML Data Server, XML queries can be created in SAP MII to retrieve a XML dataset from an URL or a file path in the SAP MII server.

The Xacute Connector

This is an internal system connector used to execute business logic transactions as data queries created using the SAP xMII Workbench, which are called *Xacute queries*. There is a default Xacute Data Server configuration available along with standard installation of SAP MII, named XacuteConnector. Xacute queries can be created using the XacuteConnector Data Server. See Figure 3-17.



Name: XacuteConnector

Connector: XacuteConnector

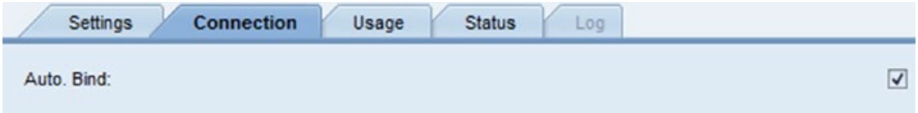
Connector Type: XCT

Description: Illuminator Xacute

Enabled: ☑

Figure 3-17. *The Xacute Connector Settings screen*

From the Connection tab of the Xacute connector, you can see Auto.Bind (see Figure 3-18). This is by default always enabled for the connector. This feature enables the values of the standard SAP MII illuminator services parameters—like IllumLoginName, IllumLoginRole, etc.—to be auto-bound from the session login.



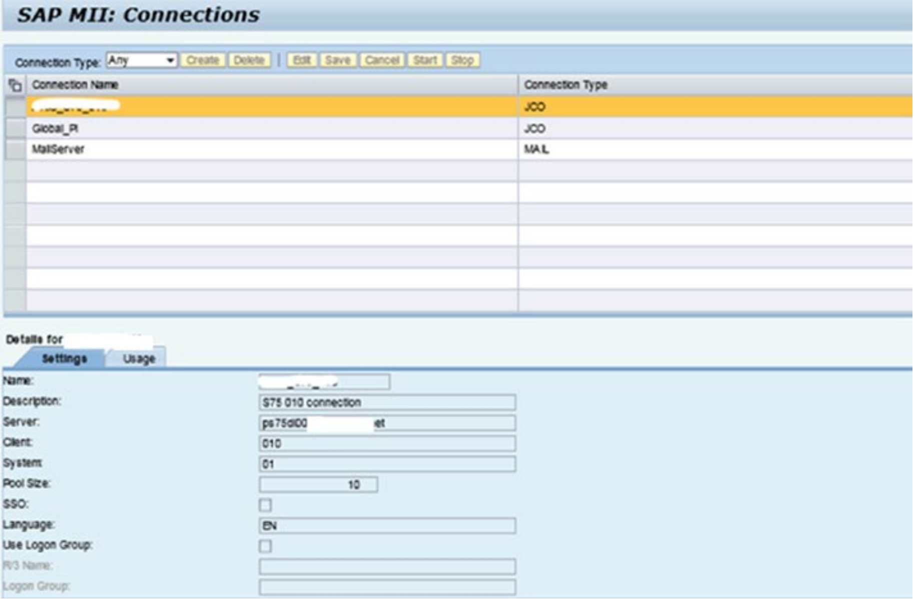
Auto. Bind: ☑

Figure 3-18. *The Xacute Connection configuration*

Connections

Connections are used to create and configure connectivity to other systems. The username and password aliases are defined in the Credential Stores screen and those aliases need to be further assigned during connection configuration. Other than this, the connection specific details like server, port, host, etc. need to be configured. Connection and credential aliases can be further used in the SAP MII action blocks, such as JCo, JRA, Send Mail, etc., instead of providing each configuration details again.

One major benefit of maintaining these aliases is if the connection details are changed, those changes can be directly incorporated into the configurations rather than touching the SAP MII codebase. In production, changing the code again and again is risky, more costly and time consuming, as it involves good amount of testing, documenting, and going through the approval process. See Figure 3-19.



Connection Type: Any

Connection Name	Connection Type
[redacted]	JCO
Global_PI	JCO
MailServer	MAIL

Details for [redacted]

Name:	[redacted]
Description:	S75 010 connection
Server:	ps75di00 [redacted] et
Client:	010
System:	01
Pool Size:	10
SSO:	□
Language:	EN
Use Logon Group:	□
R/3 Name:	
Logon Group:	

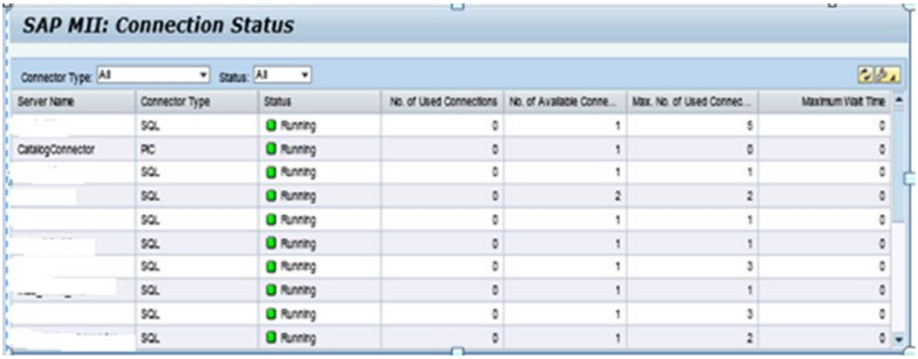
Figure 3-19. *The SAP MII Connections configuration*

The following connection types are possible:

- BC(SAP Business Connector)
- WAS (SAP NetWeaver Application Server)
- JCO (SAP Java Connector)
- FTP (File Transfer Protocol)
- MAIL (E-mail)
- JMS (Java Message Service)
- EJB (Enterprise JavaBeans)

Connection Status

The Connection Status screen displays the status of your enabled database servers. The screen only shows information for IDBC servers that have a connection pool. Most tag-based servers, such as Universal Data Servers (UDS), do not respond to a status check. See Figure 3-20.



SAP MII: Connection Status

Connector Type: All Status: All

Server Name	Connector Type	Status	No. of Used Connections	No. of Available Conne...	Max. No. of Used Connec...	Maximum Wait Time
	SQL	🟩 Running	0	1	5	0
CatalogConnector	PIC	🟩 Running	0	1	0	0
	SQL	🟩 Running	0	1	1	0
	SQL	🟩 Running	0	2	2	0
	SQL	🟩 Running	0	1	1	0
	SQL	🟩 Running	0	1	1	0
	SQL	🟩 Running	0	1	3	0
	SQL	🟩 Running	0	1	1	0
	SQL	🟩 Running	0	1	3	0
	SQL	🟩 Running	0	1	2	0

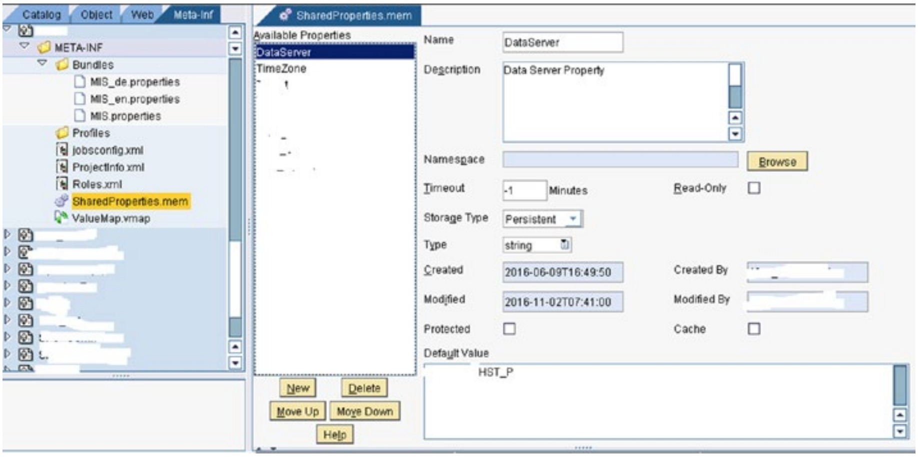
Figure 3-20. *The SAP MII Connection Status screen*

The status gets updated accordingly as and when a server has zero available connections or has a configuration, network, or communications problem that prevents SAP MII from connecting to the server.

Shared Memory

Shared memory is one of the stronger concepts of variable handling in SAP MII. It's used to store values with some properties that can be retained later. Values can be changed and deleted later. Shared properties are considered one of the temporary data storage properties. Shared properties store the value for one project across all transactions. They work like global data placeholders for particular projects.

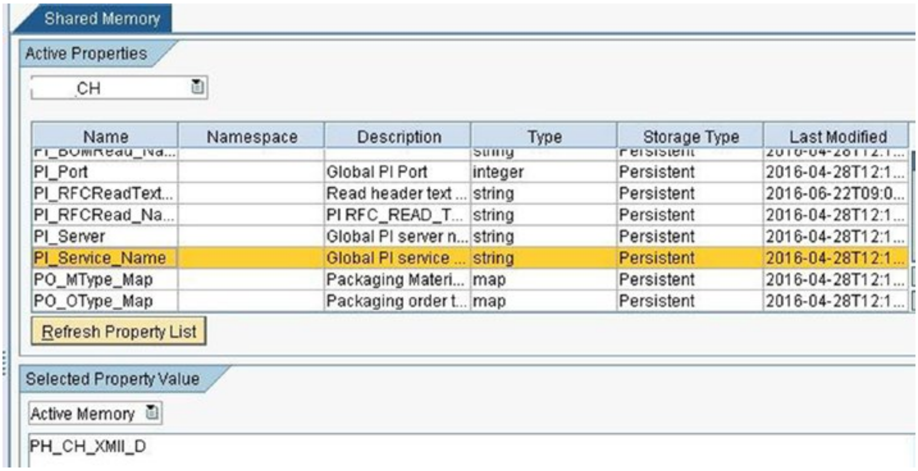
Shared properties consist of two parts: default memory and active memory. All the values are stored in the default memory when the variable is created. The same value is copied to the active memory simultaneously. The active memory is always given precedence over the default memory, as the default memory is only used so the variable has some default value. All the business logic accepts the data only from the active memory and it is available to the technical as well as admin user to change if it is required after the copied default value. You must change the default value of a shared property manually in the SAP MII Workbench. Shared properties can be defined in the `SharedProperties.mem` file of the Meta-inf tab on the SAP MII Workbench.



Name: DataServer
Description: Data Server Property
Namespace:
Timeout: -1 Minutes
Read-Only: □
Storage Type: Persistent
Type: string
Created: 2016-06-09T16:49:50
Created By: [redacted]
Modified: 2016-11-02T07:41:00
Modified By: [redacted]
Protected: □
Cache: □
Default Value: HST_P

Figure 3-21. *The Default Shared Memory configuration*

The active memory value can be changed by going to Tools ➤ Shared Memory from the SAP MII Workbench, as shown in Figure 3-22.



Name	Namespace	Description	Type	Storage Type	Last Modified
PI_BOMRead_Na...			string	Persistent	2016-04-28T12:1...
PI_Port		Global PI Port	integer	Persistent	2016-04-28T12:1...
PI_RFCReadText...		Read header text ...	string	Persistent	2016-06-22T09:0...
PI_RFCRead_Na...		PI RFC_READ_T...	string	Persistent	2016-04-28T12:1...
PI_Server		Global PI server n...	string	Persistent	2016-04-28T12:1...
PI_Service_Name		Global PI service ...	string	Persistent	2016-04-28T12:1...
PO_MType_Map		Packaging Materi...	map	Persistent	2016-04-28T12:1...
PO_OType_Map		Packaging order t...	map	Persistent	2016-04-28T12:1...

PH_CH_XMII_D

Figure 3-22. *The Shared Memory active memory configuration*

The active memory value can also be changed by going to Data Services ➤ Shared Memory of the SAP MII Menu.jsp page.

The Message Services Feature

The Message Service feature integrates enterprise systems with SAP MII. It can be used in SAP MII for receiving enterprise data in the form of asynchronous RFC, ALE/IDoc, or XML messages using HTTP POST. Enterprise systems can also send data. Message listeners can be configured in SAP MII to receive this data. Processing rules then process the data using particular business logic or messages. Messages received by MII using Message Services are stored in the SAP MI database or simply assigned to a category. These messages can be sent to external systems by MII using data services or by compiling them from external systems using web service calls in BLS. Data buffering is also done via Message Listener.

The Message Services on the SAP MII menu page have the features shown in Figure 3-23. These features are discussed in more detail in the following sections.

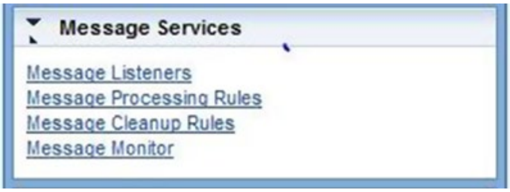
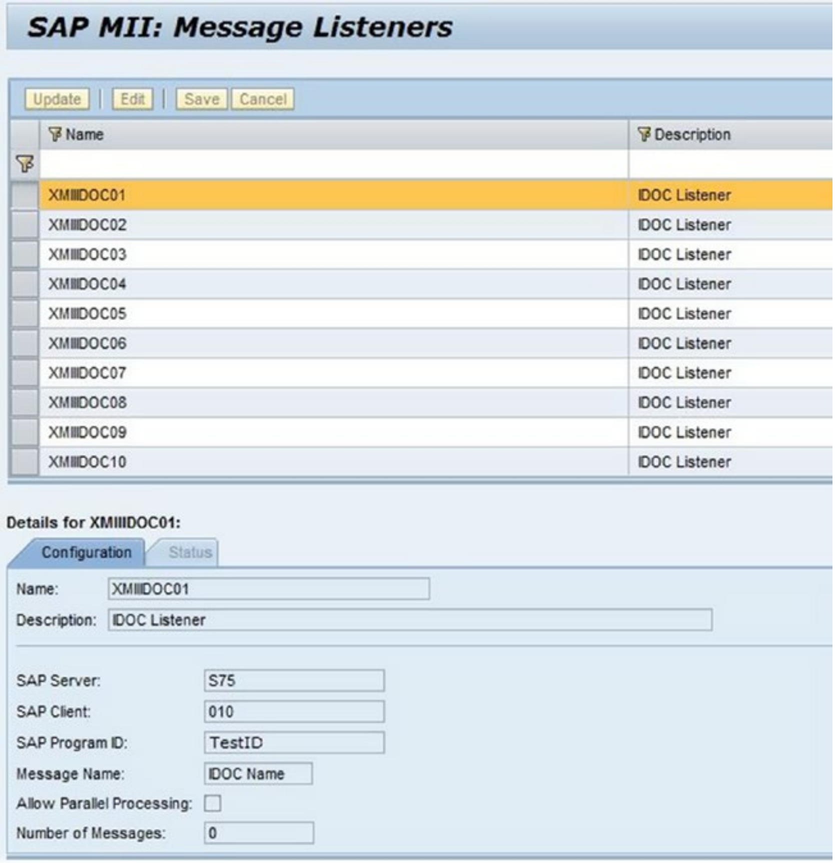


Figure 3-23. *The Message Services menu in SAP MII*

Message Listeners

Message listeners are used to receive web services, RFC (synchronous and asynchronous), or IDOC messages from enterprise systems asynchronously.

To configure an RFC/IDOC (SAPJCo) message listener, click on the Update button in the Message Listeners screen. The selected listener will be updated as per the configuration in NetWeaver. SAP MII Message Listener has three types: IDOC Listeners (10 numbers), RFC Listeners (10 numbers), and SRFC Listeners (10 numbers). See Figure 3-24.



SAP MII: Message Listeners

Name	Description
XMIIIDOC01	IDOC Listener
XMIIIDOC02	IDOC Listener
XMIIIDOC03	IDOC Listener
XMIIIDOC04	IDOC Listener
XMIIIDOC05	IDOC Listener
XMIIIDOC06	IDOC Listener
XMIIIDOC07	IDOC Listener
XMIIIDOC08	IDOC Listener
XMIIIDOC09	IDOC Listener
XMIIIDOC10	IDOC Listener

Details for XMIIIDOC01:

Name:	XMIIIDOC01
Description:	IDOC Listener
SAP Server:	S75
SAP Client:	010
SAP Program ID:	TestID
Message Name:	IDOC Name
Allow Parallel Processing:	□
Number of Messages:	0

Figure 3-24. *The Message Listener configuration in SAP MII*

To complete the Message Listener configuration for IDOC or RFC, it needs to be configured in NetWeaver. The following mandatory properties should be set by the NetWeaver server administrator:

- *ProgramID*: Unique ID and the same as the ID maintained in SAP ERP.
- *MaxReaderThreadCount*: Maximum allowed thread count in SAP ERP.
- *UserName*: Username to log in to the profile of SAP ERP.
- *Password*: User password to connect to the profile in SAP ERP.
- *ServerName*: Server URL used in profile creation in SAP ERP.
- *PortNumber*: Port number used in profile creation in SAP ERP.
- *SAPClient*: Client number used in profile creation in SAP ERP.

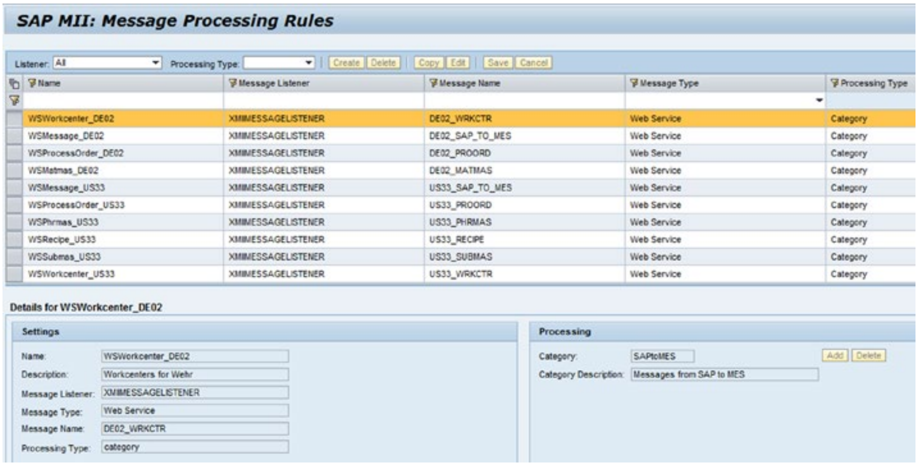
The following illuminator service is used for the message listener:
`-http://<server>:<port>/XMII/Illuminator?service=WSMessageListener&mode=WSMessageListenerServer&NAME=<UniqueMessageName>`

Specify the unique name of the General Properties section. Unchecking the Enabled checkbox allows the user to define a message listener configuration, but not activate it for use.

Any message listener configuration can be enabled or disabled as and when required. The Status property displays the current state of the listener. When a new listener is configured, the default status is always Stopped. In addition, an existing listener can be started or stopped at any point in time. A stopped message listener cannot receive a message from the sender systems.

The Message Processing Rules

The message processing rules are present to determine how a user wants to process incoming remote function calls (RFC), IDocs, and web service messages. Messages are processed based on their type. Processing rules in Message Services are defined to process messages automatically when received by SAP MII from an external sender system. See Figure 3-25.



SAP MII: Message Processing Rules

Listener: All

Processing Type:

Name	Message Listener	Message Name	Message Type	Processing Type
WSWorkcenter_DE02	XMIIMESSAGELISTENER	DE02_WRKCTR	Web Service	Category
WSMessage_DE02	XMIIMESSAGELISTENER	DE02_SAP_TO_MES	Web Service	Category
WSProcessOrder_DE02	XMIIMESSAGELISTENER	DE02_PROORD	Web Service	Category
WSMatmas_DE02	XMIIMESSAGELISTENER	DE02_MATMAS	Web Service	Category
WSMessage_US33	XMIIMESSAGELISTENER	US33_SAP_TO_MES	Web Service	Category
WSProcessOrder_US33	XMIIMESSAGELISTENER	US33_PROORD	Web Service	Category
WSPhrmas_US33	XMIIMESSAGELISTENER	US33_PHRMAS	Web Service	Category
WSRecipe_US33	XMIIMESSAGELISTENER	US33_RECIPE	Web Service	Category
WSSubmas_US33	XMIIMESSAGELISTENER	US33_SUBMAS	Web Service	Category
WSWorkcenter_US33	XMIIMESSAGELISTENER	US33_WRKCTR	Web Service	Category

Details for WSWorkcenter_DE02

Settings

Name: WSWorkcenter_DE02
Description: Workcenters for Wehr
Message Listener: XMIIMESSAGELISTENER
Message Type: Web Service
Message Name: DE02_WRKCTR
Processing Type: category

Processing

Category: SAPtoMES
Category Description: Messages from SAP to MES

Figure 3-25. *The Message Processing Rules screen*

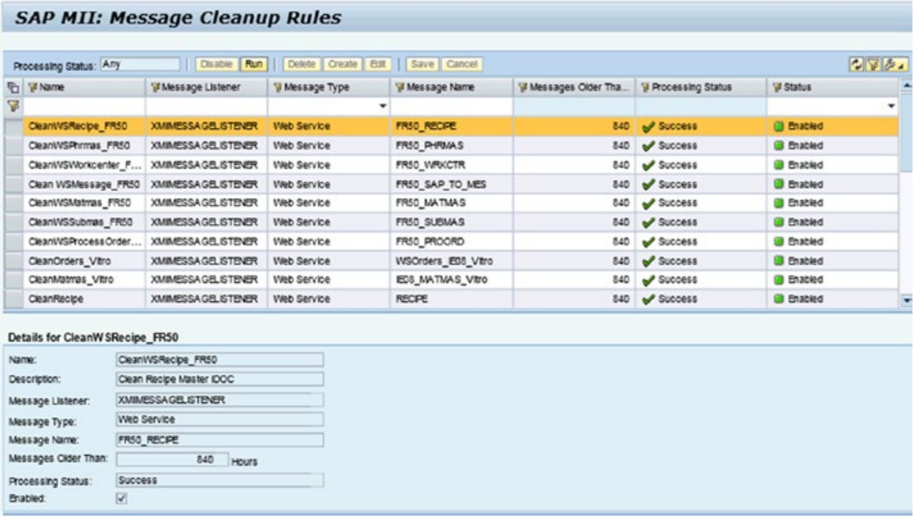
The Processing Type determines how messages are processed.

- *Transaction based rule*: Messages are immediately passed to a transaction. A business logic transaction is specified as the processing rule.

- *Category based rule*: Only the message is assigned to a category and no automatic processing is done. That is, when a specific message is received by the message listener, it is buffered into the database and the corresponding processing rule is searched. The category is used to logically group messages for ad hoc processing.

The Message Cleanup Rules

This is configured to delete buffered messages based on various conditions. The Message Cleanup screen shows a list of message cleanup rules, where you can run message cleanup rules, enable or disable message cleanup rules, and delete them as required. See Figure 3-26.



Processing Status: Any

Name	Message Listener	Message Type	Message Name	Messages Older Tha...	Processing Status	Status
CleanWSRecipe_FR50	XMIMESSAGELISTENER	Web Service	FR50_RECIPE	840	✓ Success	● Enabled
CleanWSPhrmas_FR50	XMIMESSAGELISTENER	Web Service	FR50_PHRMAS	840	✓ Success	● Enabled
CleanWSWorkcenter_F...	XMIMESSAGELISTENER	Web Service	FR50_WRKCTR	840	✓ Success	● Enabled
Clean WSMessage_FR50	XMIMESSAGELISTENER	Web Service	FR50_SAP_TO_MES	840	✓ Success	● Enabled
CleanWSMatmas_FR50	XMIMESSAGELISTENER	Web Service	FR50_MATMAS	840	✓ Success	● Enabled
CleanWSSubmas_FR50	XMIMESSAGELISTENER	Web Service	FR50_SUBMAS	840	✓ Success	● Enabled
CleanWSProcessOrder...	XMIMESSAGELISTENER	Web Service	FR50_PROORD	840	✓ Success	● Enabled
CleanOrders_Vitro	XMIMESSAGELISTENER	Web Service	WSOrders_IE08_Vitro	840	✓ Success	● Enabled
CleanMatmas_Vitro	XMIMESSAGELISTENER	Web Service	IE08_MATMAS_Vitro	840	✓ Success	● Enabled
CleanRecipe	XMIMESSAGELISTENER	Web Service	RECIPE	840	✓ Success	● Enabled

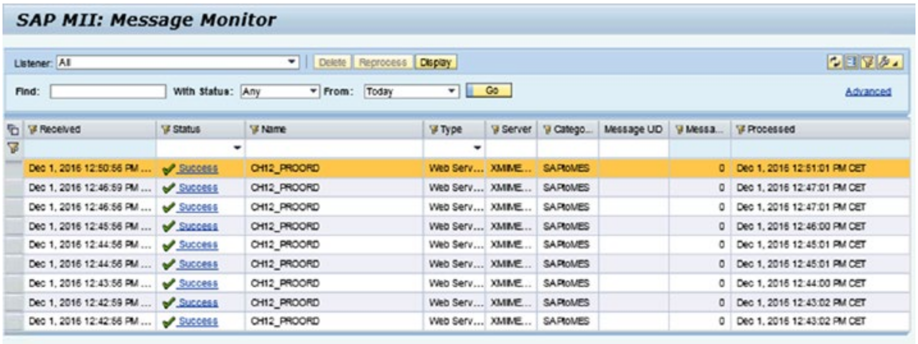
Details for CleanWSRecipe_FR50

Name: CleanWSRecipe_FR50
Description: Clean Recipe Master IDOC
Message Listener: XMIMESSAGELISTENER
Message Type: Web Service
Message Name: FR50_RECIPE
Messages Older Than: 840 Hours
Processing Status: Success
Enabled: ☑

Figure 3-26. *The Message Clean Up Rule configuration screen*

The Message Monitor Feature

The Message Monitor feature enables you to search for messages received by the message listener services, view message details, reprocess the messages, and delete messages from the databases, if needed. See Figure 3-27.



Listener: All
Find:
With Status: Any
From: Today

Received	Status	Name	Type	Server	Catego...	Message UID	Messa...	Processed
Dec 1, 2016 12:50:56 PM ...	✓ Success	CH12_PROORD	Web Serv...	XMIME...	SAPtoMES		0	Dec 1, 2016 12:51:01 PM CET
Dec 1, 2016 12:46:59 PM ...	✓ Success	CH12_PROORD	Web Serv...	XMIME...	SAPtoMES		0	Dec 1, 2016 12:47:01 PM CET
Dec 1, 2016 12:46:56 PM ...	✓ Success	CH12_PROORD	Web Serv...	XMIME...	SAPtoMES		0	Dec 1, 2016 12:47:01 PM CET
Dec 1, 2016 12:45:56 PM ...	✓ Success	CH12_PROORD	Web Serv...	XMIME...	SAPtoMES		0	Dec 1, 2016 12:46:00 PM CET
Dec 1, 2016 12:44:56 PM ...	✓ Success	CH12_PROORD	Web Serv...	XMIME...	SAPtoMES		0	Dec 1, 2016 12:45:01 PM CET
Dec 1, 2016 12:44:56 PM ...	✓ Success	CH12_PROORD	Web Serv...	XMIME...	SAPtoMES		0	Dec 1, 2016 12:45:01 PM CET
Dec 1, 2016 12:43:56 PM ...	✓ Success	CH12_PROORD	Web Serv...	XMIME...	SAPtoMES		0	Dec 1, 2016 12:44:00 PM CET
Dec 1, 2016 12:42:59 PM ...	✓ Success	CH12_PROORD	Web Serv...	XMIME...	SAPtoMES		0	Dec 1, 2016 12:43:02 PM CET
Dec 1, 2016 12:42:56 PM ...	✓ Success	CH12_PROORD	Web Serv...	XMIME...	SAPtoMES		0	Dec 1, 2016 12:43:02 PM CET

Figure 3-27. *The Message Monitor in SAP MII*

Message Services: Quality of Services

This function can be used to provide quality of service for clients in plant systems, SAP ERP, and PI. Messages can be processed in two ways.

Exactly Once (EO)

Exactly Once processes a message just one time. For RFC and IDocs, messages with the same transaction ID are rejected from message processing and then the message is stored as a duplicate in the MII system.

For HTTP messages, messageUID is used to determine if the message ID is a duplicate. The messageUID can be passed in an URL or can be created using the XPATH. The complete URL for EO support is as follows:

```
http://<server>:<port>/XMII/Illuminator?service=WSMessageListener&mode=WSMess
ageListenerServer&NAME=<UniqueMessageName>&MESSAGEUID=<UniqueMessageUID>
```

Exactly Once In Order (EOIO)

Exactly Once in Order processes messages sequentially exactly once in a queue. SAP MII processes the messages based on the sequence of the messages as sent by the client or the sender system.

Sequence numbers are sent along with the message to SAP MII. To process messages in EOIO format, EO must be supported that is, messageUID must exist.

The complete URL with sequence and message numbers is:

```
http://<server>:<port>/XMII/Illuminator?service=WSMessageListener&mode=WSMes
sageListenerServer&NAME=<UniqueMessageName>&MESSAGEUID=<UniqueMessageUID>&SE
QUENCENAME=<SequenceName>&MESSAGENUMBER=<MessageNumber>
```

Content Development

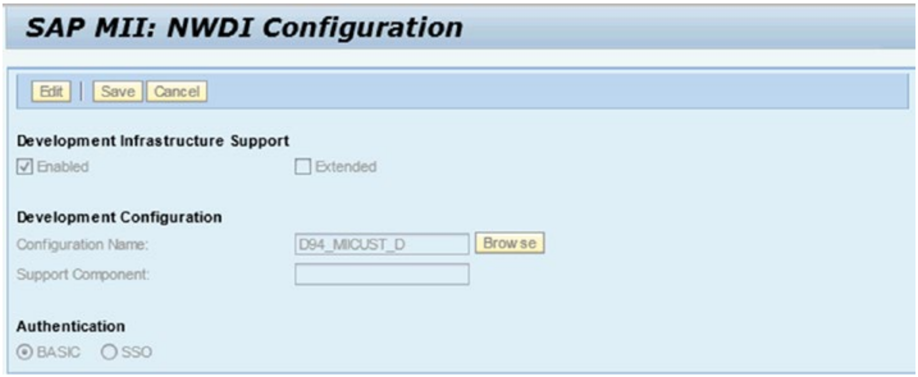
Content development is the main feature of SAP MII in which developer can develop the SAP MII solution. This section of the menu consists the IDEs for SAP MII. The following components of this menu are discussed in this section:

- NWDI Configuration
- Workbench
- Tools & Archive
- Self Service Composition Environment

NWDI Configuration

In the Content Development ➤ NWDI Configuration menu, specify the Development Track name present in NWDI to be used for the SAP MII development objects. All Workbench projects are created as Development Components (DC) in the same development track under a specific Software Component Version (SWCV).

The Development Track and SWCV are created in the DTR in NWDI, as shown in Figure 3-28.



SAP MII: NWDI Configuration

Development Infrastructure Support

☑ Enabled □ Extended

Development Configuration

Configuration Name: D94_MIICUST_D

Support Component:

Authentication

● BASIC ○ SSO

Figure 3-28. *The NWDI Configuration*

■ **Note** The DTR of the NWDI provides a versioned storage of the source files in a database. As part of the NWDI, the DTR runs as a service on the AS Java. It is seamlessly integrated into all development processes: as a developer, you can access it using the SAP NetWeaver Developer Studio, and the Component Build Service (CBS) retrieves the source files directly from the DTR. The objects in the DTR are mainly managed directly in the Change Management Service (CMS).

To be able to develop your own application, developers need to create their own software component (SC) in the System Landscape Directory (SLD). Developers have to do this before starting to develop their software. The software component is the standard delivery unit for application. The Configuration Wizard creates a default SC as a post-installation action. If a different SC is required or if the Configuration Wizard does not perform this step, a system administrator or quality manager creates the software components (SCs).

During development work, the developers add new development components to this SC.

Development Components (DCs) are the reusable building blocks for software components. A DC is a named container for arbitrary objects, e.g. Java source files, JSPs, dictionary definitions, deployment descriptors, etc. These contained objects are not directly visible to other components, i.e. the DC acts as a black box. The developers have to explicitly define "public parts" if they want some of these objects to be visible from the outside. A public part has a name and a purpose and contains a list of objects called *entities*.

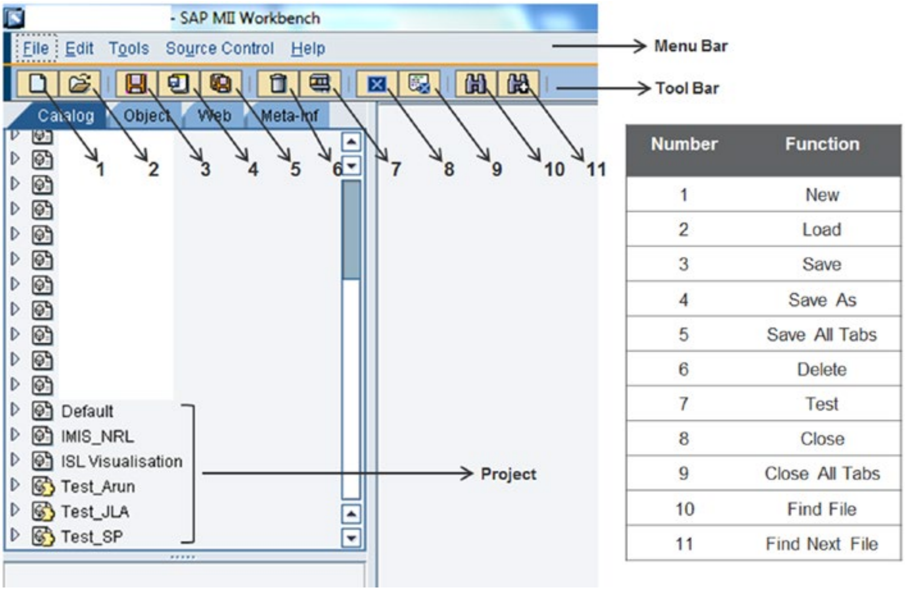
The SAP MII Workbench

SAP MII Workbench is an IDE (Integrated Development environment) for developing and managing SAP MII development objects.

In Workbench, multiple MII related objects can be developed in their respective sections. You can develop these objects in Workbench:

- BLS (Business Logic Service)
- Query template
- Display template
- MDO object
- Web page

The SAP MII Workbench has a few sections in it. Let's go through it for a better understanding. Figure 3-29 shows the full SAP MII Workbench with callouts marking its many functions.



Number	Function
1	New
2	Load
3	Save
4	Save As
5	Save All Tabs
6	Delete
7	Test
8	Close
9	Close All Tabs
10	Find File
11	Find Next File

Figure 3-29. *The SAP MII Workbench*

At the top of the IDE, there is a menu bar, which enables you to navigate throughout the IDE. Apart from that menu, there are a few standard features under Tools, like global variable declaration and active shared memory handling.

Just below the menu bar, one more bar is available. It helps you create new objects and templates as well as save those components. Apart from that, you can execute the developed content to do a sanity check or unit test while developing.

Below the bar, on the left side, there is one section consisting of four tabs. This is actually the placeholder of developmental components. It shows the hierarchical structures of the development components.

The first tab called Catalog is where business logic, query template, and display template information can be saved.

The second tab is Object and it's where developers can create their own objects for MDO, KPI, and Alert. Developers can define their object models and specifications in this section based on the development component.

The third tab is Web. This is where web coding, like HTML, IRPT, UI5, HTML5, JavaScript, and CSS, is stored. Apart from the UI components, txt, image, pdf, and XML files can also be stored on the Web tab.

The last tab is called Meta-Inf. This is where key configuration can be saved, like shared memory, project specific shared memory, roles, etc. In other words, it stores the project's metadata.

On the top-left side of SAP MII server, you see all the four tabs. Below that server, one hierarchy is maintained based on certain predefined layers. There should be one project that needs to be created under the server before starting the development. This project is the unique placeholder or logical container, known as the Development Component (DC) for the development elements for an entire solution. In SAP MII, it is possible to create two types of projects—Local projects and Shared projects.

Local projects are created and stored locally. No version controlling is available for that.

If NWDI is connected to SAP MII, it is possible to create a shared project. Basically, when a project is created in the Workbench, it will create a DC (Development Component) for the software component that's already created in NetWeaver for the specific SAP MII.

The left-bottom section is used for object-specific configuration. It populates only when some objects are selected.

The entre right section is for modeling the solution.

Business Logic Services (BLS)

Business logic services, commonly known as BLS in SAP MII, is a IDE (Interactive development environment) provided to define, design, and develop manufacturing logic. A BLS consists of a variety of action blocks, web services, a scheduler, data connectors, a logic editor, and more.

BLS transactions are like MII transactions that are graphically modeled and executable and are stored as XML in SAP MII. The internal MII content is executed at runtime by the BLS engine. See Figure 3-30.

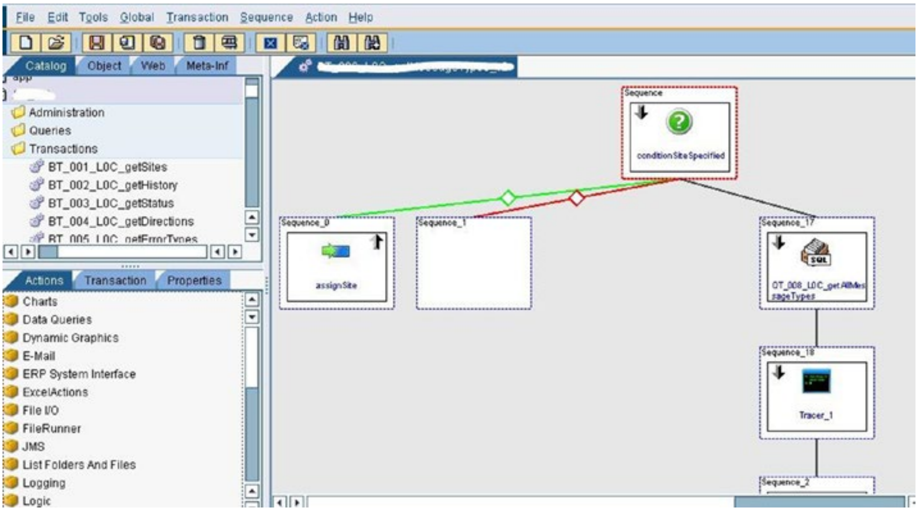


Figure 3-30. *Business Logic Services in SAP MII*

Together, these functionalities provide all of the necessary tools for defining and running business logic associated with customer-specific applications. SAP MII Business Logic Service's broad range of functionality means it can be applied in far-ranging applications, including:

- Key performance indicator (KPI) handling
- Alerts and events generation
- Data extract/transform/load (ETL)
- Application-to-application integration, including two-way ERP integration
- Dynamic content delivery of reports, graphics, and dashboard images
- Complex manipulation and transformation of data from multiple sources
- Relating and combining data from multiple sources
- Operational workflows
- Automated or manual data collection
- Production execution: tracking, routing, and monitoring

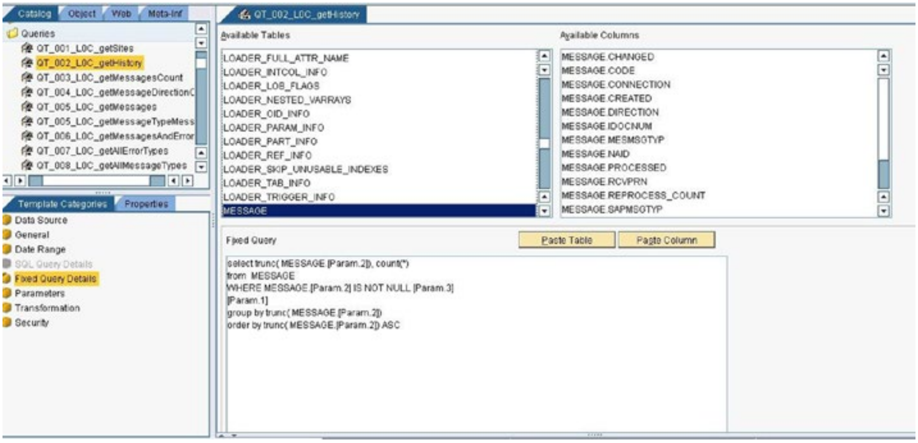
Query Template

This is the main data querying component in SAP MII. SAP MII Workbench has a few templates to create and manage data queries. By configuring these templates, it is possible to retrieve, insert, delete, and update the data sources configured in the data server sections. Query templates can be used in MII transactions; you can also call the query directly from the web pages.

Here is the list of query templates supported by SAP MII:

- `AggregateQuery`
- `AlarmQuery`
- `OLAPQuery`
- `PCoQuery`
- `MDOQuery`
- `SQLQuery`
- `TagQuery`
- `XacuteQuery`
- `XMLQuery`
- `CatalogQuery`
- `KPIQuery`

Among them, the mostly used queries are SQL Query, MDO Query, PCo Query, and Tag Query to fetch the data from disparate external source systems except MDO. Xacute Query and XML Query are the most used options for internal queries. Xacute Query is used to call transactions on the web page. XML query is used to handle XML related calls and illuminator service calls from transaction. See Figure 3-31.



QT_001_LOC_getSites
QT_002_LOC_getHistory
QT_003_LOC_getMessagesCount
QT_004_LOC_getMessageDirectionC
QT_005_LOC_getMessages
QT_005_LOC_getMessageTypeMess
QT_006_LOC_getMessagesAndError
QT_007_LOC_getAllErrorTypes
QT_008_LOC_getAllMessageTypes

LOADER_FULL_ATTR_NAME
LOADER_INTCOL_INFO
LOADER_LOB_FLAGS
LOADER_NESTED_VARRAYS
LOADER_OID_INFO
LOADER_PARAM_INFO
LOADER_PART_INFO
LOADER_REF_INFO
LOADER_SKIP_UNUSABLE_INDEXES
LOADER_TAB_INFO
LOADER_TRIGGER_INFO
MESSAGE

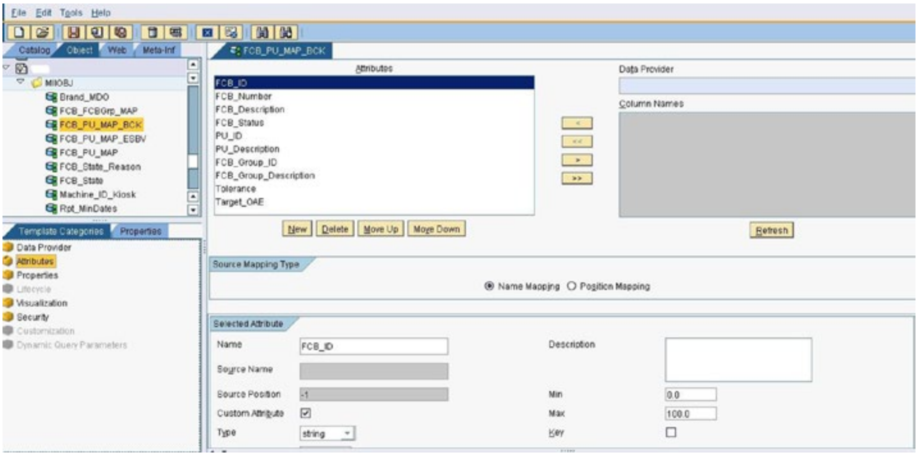
MESSAGE.CHANGED
MESSAGE.CODE
MESSAGE.CONNECTION
MESSAGE.CREATED
MESSAGE.DIRECTION
MESSAGE.IDOCNUM
MESSAGE.MESMSGTYP
MESSAGE.NAID
MESSAGE.PROCESSED
MESSAGE.RCVPRN
MESSAGE.REPROCESS_COUNT
MESSAGE.SAPMSGTYP

```
select trunc( MESSAGE.[Param.2]), count(*)
from  MESSAGE
WHERE MESSAGE.[Param.2] IS NOT NULL [Param.3]
[Param.1]
group by trunc( MESSAGE.[Param.2])
order by trunc( MESSAGE.[Param.2]) ASC
```

Figure 3-31. *The SAP MII Query Template*

MDO Object

Manufacturing Data Object (MDO) is a newly added layer in SAP MII. It is an innovative concept introduced for data acquisition in SAP MII. MDO acts as a data storage permanently or temporarily within MII. MDO can put any data object into its namespace. It mainly uses the SAP MII standard database, which is the NetWeaver database, to store the data so it is somehow a good replacement of custom database for a small-scale project. MDO has flexibility to move with the same structure and namespace when the project is moving from one system to another. Apart from object maintenance, MDO has the data querying capability to retrieve, store, update, and delete the data from any layer of data source, from plant to ERP. See Figure 3-32.



● Name Mapping ○ Position Mapping

Name: FCB_ID
Source Name:
Source Position: -1
Custom Attribute: ☑
Type: string
Description:
Min: 0.0
Max: 100.0
Key: □

Figure 3-32. *The SAP MII MDO (Manufacturing Data Object) Object Creation screen*

There are three types of MDOs available in SAP MII. They are:

- *Persistent*: Used to store and persist the data permanently.
- *On-Demand*: It is kind of a virtual table, which exists at the time of execution instance.
- *Joined*: This is a joint of two persistent MDOs.

The Visualization Services

Visualization refers to displaying a set of data so that the data points can be easily analyzed and studied in a user friendly way. The representation of data is very interactive and intelligent through various reports, dashboards, interactive presentation apps (mobile or web), etc.

Like many other platforms, SAP MII lets users create dashboard objects and web pages that can be used in various kinds of applications.

For SAP MII, visualization services are divided into Dynamic Page Generator and Web Page services.

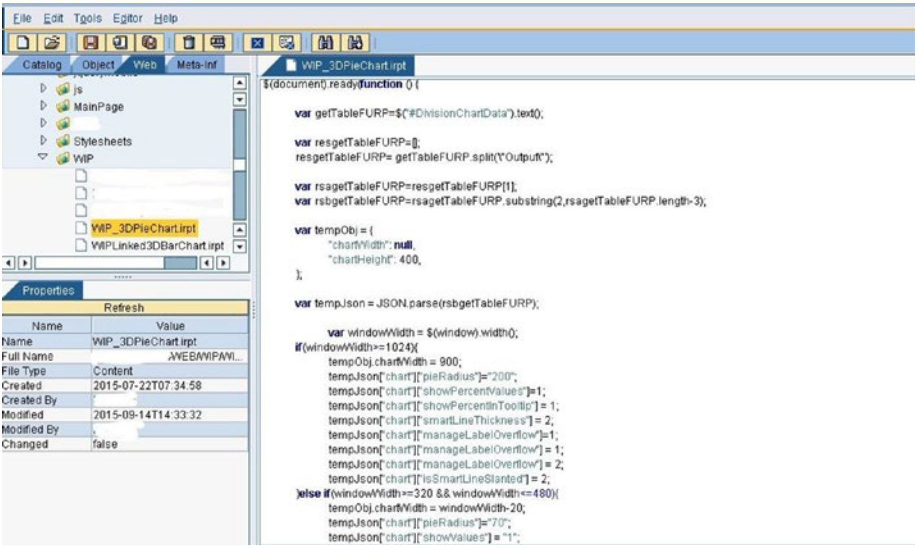
Dynamic Page Generator

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

This consists of various display template components that are available in SAP MII, like ichart, igrid, ispc, etc. For the new UI5, the display template is also available in SAP MII, like 15grid, i5chart, etc.

Web Pages

This service is used to create rich, flexible, user friendly user interface for various customer-specific dashboards. Web pages generation in SAP MII supports various UI technologies like IRPT, HTML, HTML5, UI5, etc. and supports various scripting languages like JavaScript, CSS, etc. See Figure 3-33.



Name	Value
Name	WIP_3DPieChart.irpt
Full Name	/WEB/WIP/WI...
File Type	Content
Created	2015-07-22T07:34:58
Created By	
Modified	2015-09-14T14:33:32
Modified By	
Changed	false

```
$(document).ready(function () {

        var getTableFURP=$("#DivisionChartData").text();

        var resgetTableFURP=[];
        resgetTableFURP= getTableFURP.split("\"Output\"");

        var rsagetTableFURP=resgetTableFURP[1];
        var rsbgetTableFURP=rsagetTableFURP.substring(2,rsagetTableFURP.length-3);

        var tempObj = {
                "chartWidth": null,
                "chartHeight": 400,
        };

        var tempJson = JSON.parse(rsbgetTableFURP);

                var windowWidth = $(window).width();
        if(windowWidth>=1024){
                tempObj.chartWidth = 900;
                tempJson["chart"]["pieRadius"]="200";
                tempJson["chart"]["showPercentValues"]=1;
                tempJson["chart"]["showPercentInTooltip"] = 1;
                tempJson["chart"]["smartLineThickness"] = 2;
                tempJson["chart"]["manageLabelOverflow"]=1;
                tempJson["chart"]["manageLabelOverflow"] = 1;
                tempJson["chart"]["manageLabelOverflow"] = 2;
                tempJson["chart"]["isSmartLineSlanted"] = 2;
        }else if(windowWidth>=320 && windowWidth<=480){
                tempObj.chartWidth = windowWidth-20;
                tempJson["chart"]["pieRadius"]="70";
                tempJson["chart"]["showValues"] = "1";
```

Figure 3-33. *The SAP MII Visualization through the web page service*

Animated Graphics

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

These objects are created in the SAP MII Workbench and are based on Scalable Vector Graphics (SVG), an XML-based format for representing vector drawings. See Figure 3-34.

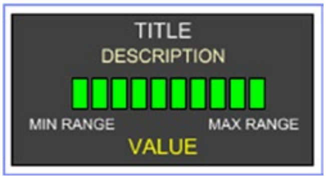


Figure 3-34. *An example of using animated graphics in SAP MII*

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

Self-Service Composition Environment

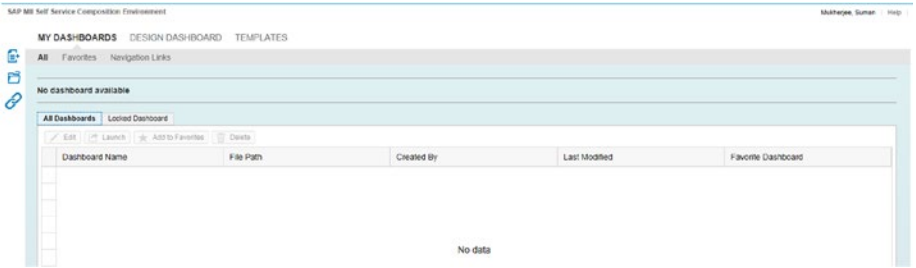
SAP MII Self-Service Composition Environment is a SAP UI5-based tool that enables users to create dashboards by dragging and dropping MII objects. It can be used by business and technical users as well.

SSCE has a powerful drag-and-drop environment in SAP MII to create dashboards using any SAP MII content (query templates, display templates, MDO/KPI objects, and resource files), UI elements, and tags from the Plant Information Catalog (PIC). Configured tags from the PIC can be positioned on images. It is possible to configure condition specific colors, graphics, and icons. It also gives you the option to create displays using different layouts of SSCE.

The Save option is also available for the created dashboard to be used later. Customization options are also available for the dashboard by adding the saved content to the navigation tree. One of the important parts of SSCE is to secure the dashboards by assigning roles.

The SAP MII Self-Service Composition Environment is divided into the following components:

- *My Dashboards Tab*: A single screen wherein users can find all their dashboards and navigation links. See Figure 3-35.



No dashboard available

Dashboard Name	File Path	Created By	Last Modified	Favorite Dashboard

No data

Figure 3-35. *The My Dashboard in SSCE*

- *Design Dashboard Tab*: A screen where the logged-in user can design/create a dashboard, as shown in Figure 3-36.
- *Left Side Toolbar*: Contains icons to trigger all the actions that could be performed in the Self-Service Composition Environment. See Figure 3-36.

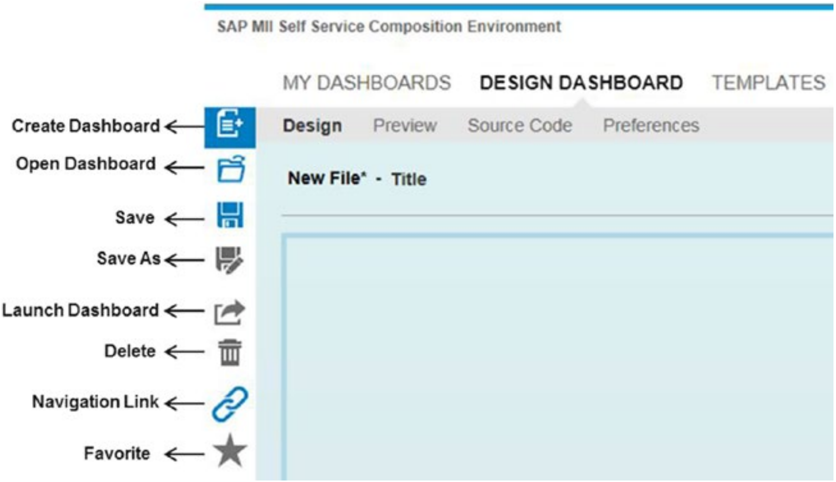


Figure 3-36. *The Design Dashboard and the left-side toolbar*

- *Right Side Pane Bar*: Contains multiple tabs, first of which is the MII objects that could be integrated with any SAP MII content like query or display template in the layouts to create dashboards. The second one is the UI elements that contain all the available UI designing controls like Button, Dropdown, Textbox, Radio button, checkbox, etc. You need to drag and drop it into the layout to design your own dashboard. Similarly, PIC and Clipboard is used to embed the PIC hierarchy and to refer to the clipboard content used in the dashboard design. See Figure 3-37.

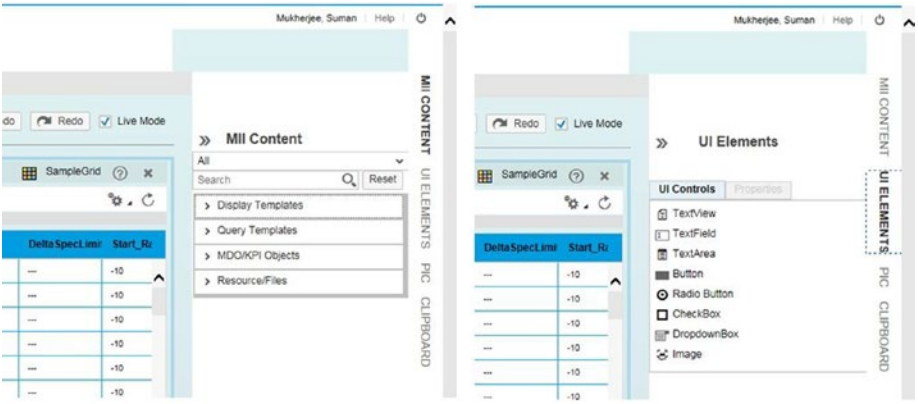
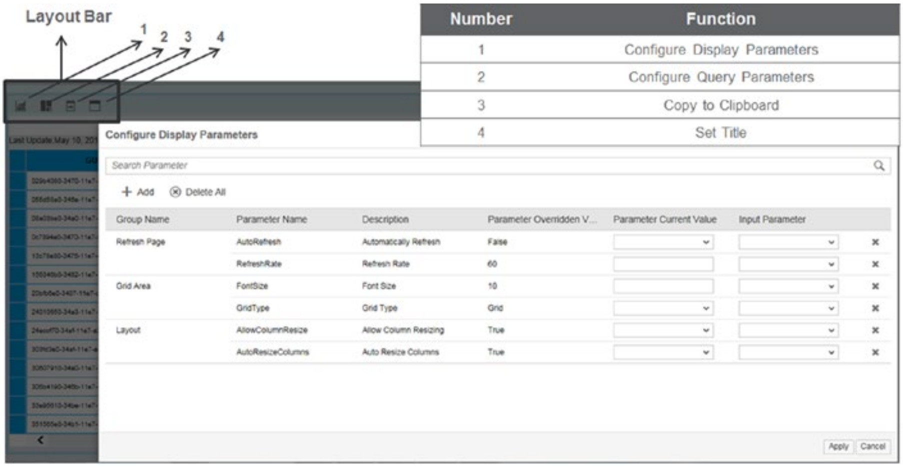


***Figure 3-37.** The Design Dashboard right side menu*

- *Layout Toolbar*: The toolbar just above the layout contains a few actions that could be performed while any MII element is embedded in the dashboard and the basic configuration needs to be changed. See Figure 3-38.



Number	Function
1	Configure Display Parameters
2	Configure Query Parameters
3	Copy to Clipboard
4	Set Title

Configure Display Parameters

Group Name	Parameter Name	Description	Parameter Overridden V...	Parameter Current Value	Input Parameter
Refresh Page	AutoRefresh	Automatically Refresh	False		
	RefreshRate	Refresh Rate	60		
Grid Area	FontSize	Font Size	10		
	GridType	Grid Type	Grid		
Layout	AllowColumnResize	Allow Column Resizing	True		
	AutoResizeColumns	Auto Resize Columns	True		

__Figure 3-38.__ The Layout bar in the Design Dashboard

Summary

In this chapter, you learned about the basic building features of SAP MII in detail, including its development configurations. In the upcoming chapters, you'll read about the functional concepts in the manufacturing industries and their associated domains.

To start with, the next chapter discussed the discrete manufacturing industry, the requirement of SAP MII in this industry, and how SAP MII provides a solution to this industry.