SAP MII – How to Browse, Configure, Test, and Consume an Enterprise Service

Applies to:

SAP MII v12.1 SAP ECC 6.0 For more information, visit the Web Services homepage.

Summary

This document will take a MII centric approach on how to browse, configure, test, and consume an Enterprise Service that is defined in ECC. The SAP MII product supports a wide range of communication protocols for interfacing with a wide range of systems in variety of ways. One of the available communication protocols is web services and in this document an end to end overview of how to utilize Enterprise Services from within MII is shown.

Author: Salvatore Castro

Web Site: http://www.scastro.net/sam

Company: SAP Labs, LLC

Created on: March 26, 2009

Author Bio

Salvatore Castro of SAP Labs and has a Bachelors Degree in Computer Engineering and a Masters Degree in Computer Science both through the Rochester Institute of Technology. He is a member of the MII Product Management group under John Schaefer.

Table of Contents

Overview

With the release of the SAP ECC 6.0 system the recommended communication protocol for retrieving enterprise data this is through the use of Enterprise Services. Currently the use of Enterprise Services requires the end user to be able to consume a web service as defined by the W3 specification. These services are designed to provide a generic interface for any application to reliably interface to the enterprise system without having to use any proprietary means of communication. The MII product has had the ability to interface to these services for multiple releases however this document is specific for v12.1 and future releases of MII. In future releases of the product it is planned to have tighter integration with the Service Repository (SR) to provide search and browse capabilities from with the MII Logic Engine environment. The MII product will also support the ability to publish a service interface to the SR to provide access to query template data via the standard web service interface.

Identifying an Enterprise Service

Depending on what you are trying to accomplish there is a wide variety of services that can be used to perform commands and query data from the ECC system. These services are defined on the SDN in the Enterprise Service Workplace here: https://www.sdn.sap.com/irj/sdn/esworkplace and more specifically for manufacturing integration here: **https://www.sdn.sap.com/irj/scn/wiki?path=/display/ESpackages/Integration+of+Manufacturing+Execution+Systems**. When identifying a service it is important to understand how the service works and the various input parameters that are required or optional for the service. As an example the enterprise service to "Find a Production Order By Elements" is used in this document. To find the documentation for this service click on the previous link and then click on "Enterprise Services in Integration of Manufacturing Execution Systems". Next from the list of possible services select "Find Production Order by Elements" and finally select the "View this enterprise service" link. The documentation for this service can be found here but it's important to know how to get to this page for the other services: http://esworkplace.sap.com/socoview(bD1lbiZjPTAwMSZkPW1pbg==)/render.asp?sap-unique=184728&sap-params=aWQ9RDUwN0MxNjE3NkJFMTFEQTM2QkIwMDBGMjBEQUM5RUYmcGFja2FnZWlkPURFMDQyNkREOUIwMjQ5RjE5NTE1MDAxQTY0RDNGNDYy.

When working with services it is important to first note the "Technical Name" or the "Internal Name" given to the service, and in this case it is: ProductionOrderSimpleByElementsQueryResponse_In (This name is used in the next section). Also refer to the Usage Notes and examples along with definitions for the various usage codes that are required (See Appendix A for details).

Defining the Endpoint and Obtaining the WSDL URL

Once you've identified the service you wish to consume the next step is to define the endpoint for the service in your ECC system. In order to accomplish this login to ECC via SAP GUI and navigate to transaction code **SOAMANAGER**. This will open up a web Dynpro based interface for enterprise service configuration and administration. Follow these steps:

1. Select the Application and Scenario Configuration tab and then select the Single Service Administration link.
2. From the Search interface enter *Prod*Order* in the Search Pattern Field, select External Name from the Field drop-down and press the Go button.
3. A list of services will appear and you can scroll down to the ProductionOrderSimpleByElementsQueryResponse_In name and select the row. Notice that the internal name for this service is ECC_PRODUCTIONORDERSEQR.
4. Press the Apply Selection button and configuration details about the service will appear.
5. From the Overview tab select the Display selected Binding's WSDL URL link and the WSDL URL that is defined for the service will appear in a text area on the right. If the text area is empty then continue onto the next step, otherwise skip to step 9.
6. Since the WSDL URL is not defined yet, one has to be configured before continuing and can be done by selecting the Configurations tab and pressing the Create Service Button.
7. A new window will appear that allows you to define the following:
    a. New Service Name = PRODORDERSIMPLEELEMENTS
    b. Description = Select a List of Production Orders based on search criteria
    c. New Binding Name = ProdorderSimpleElementsQuery
8. Press the Apply Settings button and scroll down to the Authentication Settings -> Authentication Method section. Verify that only the Transport Channel Authentication -> User ID/Password checkbox is checked. If it's not checked be sure to check it. Press the Save Button.
9. Select the Overview Tab and Press the Display selected Binding's WSDL URL link and verify that the WSDL for the service appears in the text area.

Testing the Service

When testing the enterprise services there are two mechanisms built into ECC that can be used and the third is to utilize the MII Web Service Action (In v12.2 the Service Repository Action).

Testing the Service via SAP GUI

In order to test a service via SAP GUI follow these steps:

1. Navigate to Transaction SE84
2. Select Repository Information System -> Enterprise Services -> Service Definitions
3. In the Service Definition field enter the Internal Name for the service, which in the aforementioned example is ECC_PRODUCTIONORDERSEQR
4. Press the Execute button or Press the F8 key
5. Double-Click on the service definition that appears and this will load the configuration information for the specified service.
6. Press the Test button or Press the F8 key
7. Select the testing environment you wish to use and press the Execute button or F8 key
8. A screen should appear that shows the Request XML structure generated from the WSDL definition for the service. You can press the XML Editor button at the top or Ctrl+F2 to edit the XML Request payload you wish to submit to the service.
9. Once the XML Payload is defined press the Execute button or F8 key to execute the request. The response will appear in a separate Response tab next to the Request Tab.

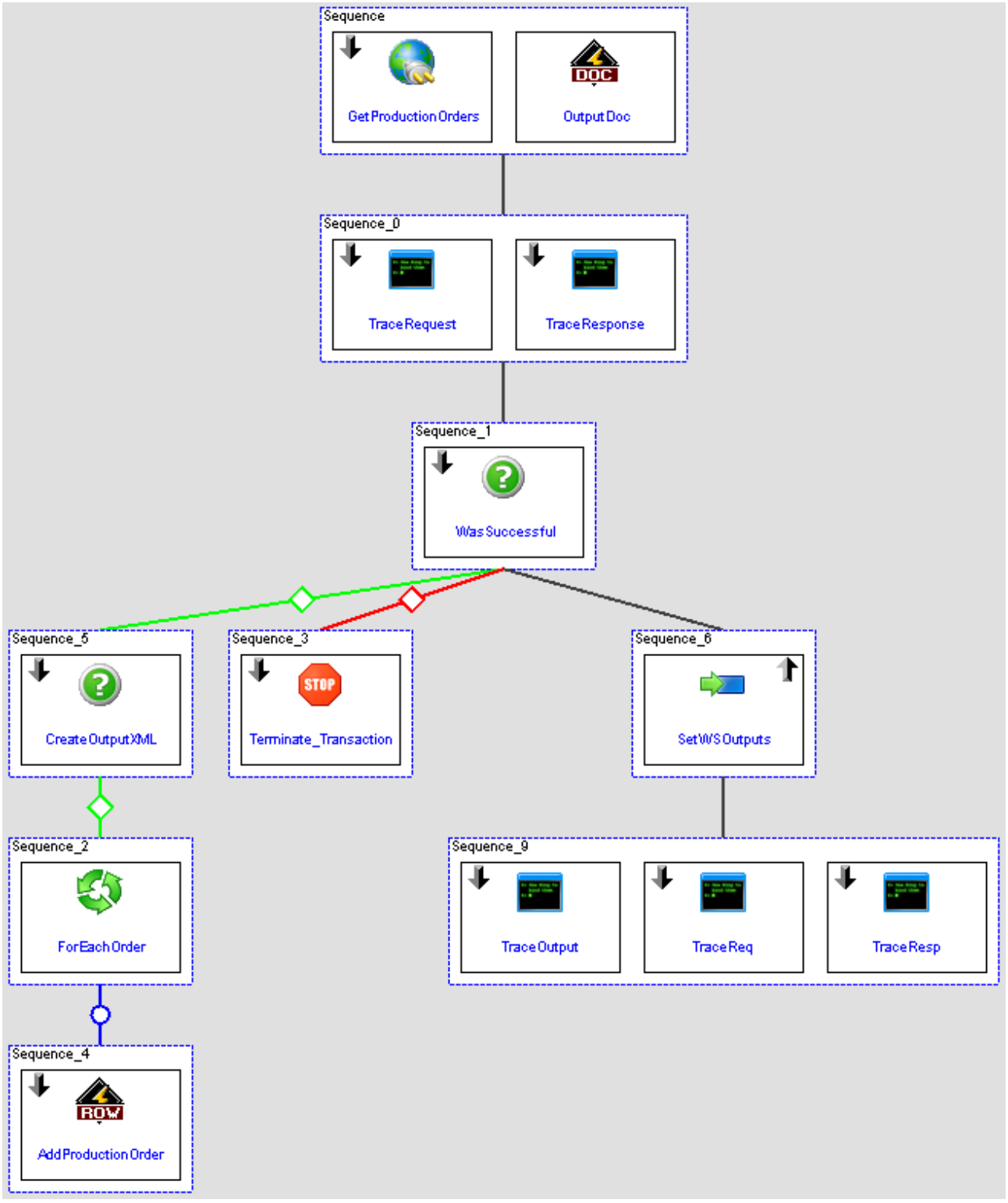
Testing the Service via SOAMANAGER

The service can also be tested via the Web Dynpro interface to the Service Repository.

1. Log into SAP GUI and navigate to transaction code SOAMANAGER
2. This will load the Web Dynpro interface previously used to defined the Service WSDL
3. Find the service you wish to use, select it and press the Apply Selection button
4. Select the Open Web Service Navigator for selected link and Log in again.
    a. The first login is for the XI/PI system where the service repository is hosted
    b. The second login is for the Web Service Request
5. Select the Operation you wish to use under the selected service and press the Next button
6. Enter in the Service details and press the Next button
7. Enter in the various values you wish to submit to the service and press Next
8. Here you can view the results of the service request execution

Utilizing the Service via MII

From the MII Menu open up the workbench via Data Services -> MII Workbench and navigate to your project. Create a New Transaction named ProductionOrderList that looks like this:



Create the following Transaction Properties:

Name	DataType	Description	Type
CreateOutputXML	Boolean	Boolean flag to generate the MII formatted XML Output	Input
EndingPlanningHorizon	Int	The number of days in the future to look for orders	Input
Material	String	The ECC defined material to lookup orders for	Input
OutputXML	XML	The MII formatted XML response from the request	Output
Plant	String	The ECC defined plant id to lookup orders for	Input
RequestXML	XML	The XML Web Service request message	Output
ResponseXML	XML	The XML Web Service response message	Output
StartingPlanningHorizon	Int	The number of days in the past to look for orders	Input

Create the following Local Properties:

Name	DataType	Description	DefaultValue
ExclusionCode	String	The exclusive search indicator	E
InclusionCode	String	The inclusive search indicator	I
IntervalBoundry_EQ	String	The equality range indicator	1
IntervalBoundry_BT	String	The range indicator	3

Configure the web service action to point to the WSDL you defined using the steps in the previous section and set the following links for the action under the ns1:ProductionOrderSimpleByElementsQuery_sync/ProductionOrderSimpleSelectionByElements path:

Target	Link Expression
SelectionByPlant/InclusionExclusionCode	Local.InclusionCode
SelectionByPlant/IntervalBoundaryTypeCoe	Local.IntervalBoundary_EQ
SelectionByPlant/LowerBoundaryPlantID	Transaction.Plant
SelectionByProductionOrderTypeCode/InclusionExclusionCode	Local.InclusionCode
SelectionByProductionOrderTypeCode/IntervalBoundaryTypeCode	Local.IntervalBoundary_EQ
SelectionByProductionOrderTypeCode/ProductionOrderTypeCode	“PP01”
SelectionByProductionOrderPlannedStartDate/InclusionExclusionCode	Local.InclusionCode
SelectionByProductionOrderPlannedStartDate/IntervalBoundaryTypeCode	Local.IntervalBoundary_BT
SelectionByProductionOrderPlannedStartDate/LowerBoundaryOrderPlannedStartDate	datefromxmlformat(dateadddays(datenow, -Transaction.StartingPlanningHorizon), "yyyy-MM-dd")
SelectionByProductionOrderPlannedStartDate/UpperBoundaryOrderPlannedStartDate	datefromxmlformat(dateadddays(datenow, Transaction.EndingPlanningHorizon), "yyyy-MM-dd")
SelectionByMaterialID/InclusionExclusionCode	Local.InclusionCode
SelectionByMaterialID/IntervalBoundaryTypeCode	Local.IntervalBoundary_EQ
SelectionByMaterialID/LowerBoundaryMaterialID	Transaction.Material
SelectionBySupplyPlanningPlantID/InclusionExclusionCode	Local.InclusionCode

SelectionBySupplyPlanningPlantID/IntervalBoundaryTypeCode	Local.IntervalBoundary_EQ
SelectionBySupplyPlanningPlantID/LowerBoundarySupplyPlanningPlantID	Transaction.Plant
ERPProcessOrderRequiredIndicator	“false”

The rest of the nodes that are not set must be removed form the request in order for the request to succeed.

Next configure the OutputDoc for the following:

Column Name	Type	Description
ProductionOrder	String	The production order number
Material	String	The material that is to be produced by the order
SalesOrder	String	Sales order that triggered the creation of the production order
Status	String	The current status of the order
PlannedStartDate	DateTime	The date when the order is planned to begin
PlannedEndDate	DateTime	The date when the order is planned to be completed by
PlannedQuantity	Double	The quantity of material to be produced
UOM	String	The unit of measure of the quantity to produce

Then in the WasSuccessful conditional action configure the action for two inputs:

Target	Link Expression
Input1	GetProductionOrders.Success
Input2	GetProductionOrders.ProductionOrderSimpleByElementsResponse_sync {/ns1:ProductionOrderSimpleByElementsResponse_sync/Log/Item/TypeID} != "E"

Configure the Repeater to repeat on the following:

GetProductionOrders.ProductionOrderSimpleByElementsResponse_sync{/ns1:ProductionOrderSimpleByElementsResponse_sync/ProductionOrder}

Map in the parameters from the Row action block to configured the OutputDoc columns.

The Terminate_Transaction action block should have the following linked to the Message:

GetProductionOrders.ProductionOrderSimpleByElementsResponse_sync{/ns1:ProductionOrderSimpleByElementsResponse_sync/Log/Item/Note}

In the SetWSOutput action link the Transaction RequestXML & ResponseXML to the XML request and response properties from the GetProductionOrders web service action and link the OutputDoc XML to the OutputXML transaction property.

Appendix A

Interval Boundary Type Codes

Code	Name	Description
1	Single Value; = X	Single Value; = X
2	Interval with clsd lower and open upper boundary; [X, Y)	Interval with clsd lower and open upper boundary; [X, Y)
3	Interval with clsd lower and clsd upper boundary; [X, Y]	Interval with clsd lower and clsd upper boundary; [X, Y]
4	Interval with open lower and open upper boundary; (X, Y)	Interval with open lower and open upper boundary; (X, Y)
5	Interval with open lower and clsd upper boundary; (X, Y]	Interval with open lower and clsd upper boundary; (X, Y]
6	Interval with unlmt lower and open upper boundary; < X	Interval with unlmt lower and open upper boundary; < X
7	Interval with unlmt lower and clsd upper boundary; <= X	Interval with unlmt lower and clsd upper boundary; <= X
8	Interval with open lower and unlmt upper boundary; > X	Interval with open lower and unlmt upper boundary; > X
9	Interval with clsd lower and unlmt upper boundary; >= X	Interval with clsd lower and unlmt upper boundary; >= X

Production Type Codes

Code	Name	Description
H40	Final Production Order Production Confirmation	Final confirmation of a production order. No further confirmations are expected.
B40	Final End of Processing	"End of Processing' time event - final confirmation
A10	Start of Teardown	"Start of Teardown' time event
A40	Final End of Teardown	"End of Teardown' time event - final confirmation
L00	Automatic Final Time Ticket Production Confirmation	Final confirmation of a step in the production process, if the confirmed actual quantity (yield + scrap) equals the planned values. If not, a partial confirmation is executed.
L40	Final Time Ticket Production Confirmation	Final confirmation of a step in the production process. No further confirmations are expected.
H00	Automatic Final Production Order Production Confirmation	Final confirmation of a production order, if the confirmed actual quantity (yield + scrap) equals the planned values. If not, a partial confirmation is executed.
A20	Partial End of Teardown	"End of Teardown' time event - partial confirmation
B20	Partial End of Processing	"End of Processing' time event - partial confirmation
B30	Interrupt of Processing	"Interruption of Processing' time event
B10	Start of Processing	"Start of Processing' time event
V40	Final Activity Production Confirmation	Final confirmation of actual work quantities for a single step in the production process. No further confirmations are expected.
R30	Interrupt of Setup	"Interruption of Setup' time event
R40	Final End of Setup	"End of Setup' time event - final confirmation
L20	Partial Time Ticket Production Confirmation	Partial confirmation of a step in the production process. Further confirmations are expected.
H20	Partial Production Order Production Confirmation	Partial confirmation of a production order. Further confirmations are expected.
R10	Start of Setup	"Start of Setup' time event
V20	Partial Activity Production Confirmation	Partial confirmation of actual work quantities of a single step in the production process. Further confirmations are expected.
A30	Interrupt of Teardown	"Interruption of Teardown' time event
R20	Partial End of Setup	"End of Setup' time event - partial confirmation

Related Content

Enterprise Services Repository & Registry Help Documentation:
http://help.sap.com/saphelp_nwpi71/helpdata/en/c7/4ce1aa448945b5bdf51566b09b86e3/content.htm

Enterprise Services Wiki: https://www.sdn.sap.com/irj/sdn/esworkplace

MES Integration Services Wiki:
https://www.sdn.sap.com/irj/scn/wiki?path=/display/ESpackages/Integration+of+Manufacturing+Execution+Systems

MII Forum: https://www.sdn.sap.com/irj/sdn/forum?forumID=237

- **Obtaining a Service WSDL URL**: https://forums.sdn.sap.com/thread.jspa?threadID=1130662

MII Wiki: https://wiki.sdn.sap.com/wiki/display/xMII

World Wide Web Consortium on WSDL: http://www.w3.org/2002/ws/desc/

Web Service Reliable Messaging: http://en.wikipedia.org/wiki/WS-ReliableMessaging

Relevant SAP Notes

960040 - Using Enterprise Services

637388 - Enterprise Services Architecture: Delivery 6.20

Copyright