How To... Develop, Monitor and Debug WS Consumer and Provider

Applicable Releases:
SAP NetWeaver 7.0 SP14

IT Practice:
Service SOA & Design

IT Scenario:
Enabling Enterprise Services

Version 1.1
March 2010
## Document History

<table>
<thead>
<tr>
<th>Document Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.10</td>
<td>Minor changes</td>
</tr>
<tr>
<td>1.00</td>
<td>First official release of this guide</td>
</tr>
</tbody>
</table>
## Typographic Conventions

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example Text</td>
<td>Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation</td>
</tr>
<tr>
<td>Example text</td>
<td>Emphasized words or phrases in body text, graphic titles, and table titles</td>
</tr>
<tr>
<td>Example text</td>
<td>File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.</td>
</tr>
<tr>
<td>Example text</td>
<td>User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.</td>
</tr>
<tr>
<td><code>&lt;Example text&gt;</code></td>
<td>Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.</td>
</tr>
<tr>
<td>EXAMPLE TEXT</td>
<td>Keys on the keyboard, for example, F2 or ENTER.</td>
</tr>
</tbody>
</table>

## Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🚨</td>
<td>Caution</td>
</tr>
<tr>
<td>📚</td>
<td>Note or Important</td>
</tr>
<tr>
<td>🕵️</td>
<td>Example</td>
</tr>
<tr>
<td>🔴</td>
<td>Recommendation or Tip</td>
</tr>
</tbody>
</table>
## Table of Contents

1. Business Scenario.................................................................................................................. 1
2. Background Information......................................................................................................... 1
3. Prerequisites .......................................................................................................................... 1
   3.1 Check and Verify WS-RM Configurations ................................................................. 1
   3.2 Configure WS-RM for Consumer and Provider System ............................................ 3
4. Step-by-Step Procedure ........................................................................................................ 13
   4.1 Create the Provider Application .................................................................................... 13
   4.2 Create the Consumer Application .................................................................................. 22
   4.3 Monitoring and Debugging .............................................................................................. 31
   4.4 Use WSNavigator to Debug Web Services ..................................................................... 40
1. **Business Scenario**

Since SAP NetWeaver 7.0, SP 14, Web Services Reliable Messaging (WS-RM) has been introduced. WS-RM provides reliable delivery of asynchronous messages using the SOAP protocol. WS-RM is part of the ABAP stack. In this how-to guide, we take a look at the development of consumer proxies and provider services using WS-RM. In our business scenario, we use a simple order creation process on a SAP NetWeaver 7.1, SP 7 system (AS ABAP). To simplify the application for demo purposes, the service inserts the order information into a database table. Then, we implement two consumer proxies on an SAP NetWeaver 7.0, SP 18 and an SAP NetWeaver 7.1, EHP 1, SP 3 system (AS ABAP). This is to show interoperability between the different releases.

In addition, we look at the monitoring of the messages sent from the consumer to the provider system.

**Note**

WS-RM protocol supports only asynchronous messaging. When interfaces are configured as synchronous using WS-RM, messages are treated as standard SOAP requests.

2. **Background Information**

The standard SOAP protocol does not guarantee the delivery of messages. With the inclusion of the WS-RM protocol, message delivery is guaranteed. When a sequence of messages is sent from the consumer systems, these messages are tracked by the provider system. If any of the messages do not arrive, the provider requests the missing message to be re-sent. Consequently, a WS-RM runtime engine needs to be available at both the consumer and provider systems. This guaranteed message delivery has already been available with SAP NetWeaver PI/XI systems. Now the same quality of service can be handled by WS-RM, without using SAP NetWeaver PI/XI.

3. **Prerequisites**

3.1 **Check and Verify WS-RM Configurations**

Before you can use WS-RM, you must verify that WS-RM has been configured on the consumer and provider systems.

1. In transaction SICF, activate the following nodes, if not already activated:
   a. /sap/bc/srt (including sub nodes)
   b. /sap/bc/webdynpro/sap/appl_soap_management
2. For each system, execute the report using SE38: SRT_ADMIN_CHECK

Example output of a not configured WS-RM system:

### Check Administration Environment of SOAP Runtime

<table>
<thead>
<tr>
<th>Checking cross-system settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>bgRFC destination is not operational</td>
</tr>
<tr>
<td>bgRFC supervisor destination is not registered</td>
</tr>
<tr>
<td>WSRM event handler is not active</td>
</tr>
<tr>
<td>Task watcher is not active</td>
</tr>
<tr>
<td>Data collector for monitoring is activated</td>
</tr>
<tr>
<td>Ending check of cross-system settings</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Checking client-specific settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service destination in client 000 is not configured</td>
</tr>
<tr>
<td>Service destination in client 001 has different Unicode settings</td>
</tr>
<tr>
<td>Service destination in client 001 is not operational</td>
</tr>
<tr>
<td>Service destination in client 006 is not configured</td>
</tr>
<tr>
<td>Service destination in client 105 does not have a valid service user</td>
</tr>
<tr>
<td>Service destination in client 105 has different Unicode settings</td>
</tr>
<tr>
<td>Service destination in client 105 is not operational</td>
</tr>
<tr>
<td>Service destination in client 106 does not have a valid service user</td>
</tr>
<tr>
<td>Service destination in client 106 has different Unicode settings</td>
</tr>
<tr>
<td>Service destination in client 106 is not operational</td>
</tr>
<tr>
<td>Service destination in client 107 does not have a valid service user</td>
</tr>
<tr>
<td>Service destination in client 107 has different Unicode settings</td>
</tr>
<tr>
<td>Service destination in client 107 is not operational</td>
</tr>
<tr>
<td>Ending check of client-specific settings</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Client-specific connection checks (destinations)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service destination in client 000 Unregistered</td>
</tr>
<tr>
<td>Connection Test in Client 001 successful</td>
</tr>
<tr>
<td>Service destination in client 066 Unregistered</td>
</tr>
<tr>
<td>Error in service destination in client: 105 Name or password is incorrect (repeat logon)</td>
</tr>
<tr>
<td>Error in service destination in client: 106 Name or password is incorrect (repeat logon)</td>
</tr>
<tr>
<td>Error in service destination in client: 107 Name or password is incorrect (repeat logon)</td>
</tr>
<tr>
<td>End of client-specific connection tests (destinations)</td>
</tr>
</tbody>
</table>
Example output of a configured WS-RM system:

### Check Administration Environment of SOAP Runtime

- Checking cross-system settings
  - bgRFC destination is operational
  - bgRFC supervisor destination is registered
  - WSRM event handler is active
  - Task watcher is active
  - Data collector for monitoring is activated
  - Ending check of cross-system settings

- Checking client-specific settings
  - Service destination in client 066 is not configured
  - Ending check of client-specific settings

- Client-specific connection checks (destinations)
  - Connection Test in Client 000 successful
  - Connection Test in Client 001 successful
  - Service destination in client 066 Unregistered
  - Connection Test in Client 100 successful
  - Connection Test in Client 105 successful
  - Connection Test in Client 106 successful
  - Connection Test in Client 107 successful
  - End of client-specific connection tests (destinations)

If the report output does not reflect similarity to a properly configured WS-RM, follow the steps in section 3.2, else, skip the next section and go to section 4.

# 3.2 Configure WS-RM for Consumer and Provider System

**Important**

Follow the steps in this section only if WS-RM has not been configured based on the instructions above.

**Note**

Information on SAP Help Portal: [Configuring the Web Service Runtime](#)
1. On all clients, use transaction SCUA to check whether CUA is active.
   If it is not active, the following screen is displayed:

   ![Maintain system landscape](image)

   - If not active, use function module (SE37) SRT_TECHNICAL_SETUP
     
     **Note**
     
     SAP Note 1110741 [Web service configuration can only be executed using SE37](#)

   - If active, use report (SE38) SRT_ADMIN
     
     **Note**
     
     SAP Note 1043195 [Configuration of the Web service runtime](#)

2. On all clients, make sure the user used to perform the configurations has SAP_ALL rights, or DDIC rights.

3. On Client 000:
   a. Execute report (SE38) SRT_ADMIN
   b. Select option *Perform Technical Setup* and execute.
When completed, the following window is displayed.

### Technical Configuration of SOAP Runtime

- **Configuration successful**

Re-execute report SRT_ADMIN, ABAP Connections and Web Service-Specific bgRFC Setting are displayed:

#### Technical Configuration of SOAP Runtime

- **ABAP Connections**
  - Register ABAP Connection
  - Name of ABAP Connection: WS_SERVICE_1D0E69932B23_SRV

- **Web Service-Specific bgRFC Settings**
  - Register Name of Inbound Destination
  - Name of Inbound Destination: WS_SERVICE_1D0E600247C4_IN

- **Runtime Trace**
  - Configure Trace
    - User Name: LTwI
    - Trace Level: 1
    - Validity [sec]: 300

- **Configure Logging**
  - Configure Logging
    - User Name: LTwI
    - Log Level: 1

**Tip**

During the configuration process, you can use report SRT_ADMIN_CHECK to verify pending and completed tasks.
4. On Client 000, configure bgRFC.
   a. Execute transaction: SBGRFCCONF
   b. On the **Scheduler App Server** tab, click on **Create**:

   ![Scheduler App Server settings](image)

   c. Provide a server name and save:

   ![Create Scheduler Settings Server](image)
d. This is the result:

No other entries are necessary.

e. Check whether the queue prefixes are registered. If the setup is successful, no entries are necessary on this tab.
f. To define an RFC destination, click *Create* on the *Define Supervisor Destination* tab and enter the necessary information:

i. Provide a password and select *Save*. The user S_PI1_000 is also created.

Check whether the user is a service user and has the role *SAP_BC_BGRFC_SUPERVISOR* assigned. (Add the role, if not present.)

The RFC destination (type 3) should also be created: BGRFC_SUPERVISOR

The RFC destination should use the logon user: S_PI1_000

ii.
iv. Save the entries:

```
bgRFC Configuration
```

g. Verify the configuration by executing the report SRT_ADMIN_CHECK. If error *No RFC authorization for function module RFCPING*, occurs, follow the steps described below. Use transaction PFCG and add the role `SAP_BC_WEB_SERVICE_SERVICE_USER` with authority for RFC function group SYST.

i. In transaction PFCG, select *Change*, select the *Authorization* tab and click on *Change Authorization Data*: 
ii. Use the screenshot below to expand the folder and click on change for Name of RFC to be protected:
iii. Enter SYST and Save:

![Field values](image)

iv. Generate the change:

![Change role: Authorizations](image)

v. To check that there are no more errors, execute report SRT_ADMIN_CHECK.
h. To configure WS-RM runtime event handler, execute report RSEHCONFIG:

![Configure Runtime Behavior of WSRM Event Handler](image)

i. Start background job BC_SAP_SOAP_RUNTIME_MANAGEMENT.
   If not scheduled in transaction SM37, call transaction SM36, and choose Standard Jobs - Standard Scheduling. Schedule the job hourly.

5. On all other clients, execute report SRT_ADMIN. Execute transaction PFCG to enhance role SAP_BC_WEBSERVICE_SERVICE_USER with authority for RFC function group SYST using the instructions previously outlined.

6. To check the configurations, execute report SRT_ADMIN_CHECK:

![Check Administration Environment of SOAP Runtime](image)
4. **Step-by-Step Procedure**

In this section, we implement WS-RM provider and consumer applications on SAP NetWeaver. The provider application is an ABAP proxy. The ABAP proxy is generated using an inbound service interface defined in the ES Repository. Once the proxy is generated, use SOA Management (transaction SOAMANAGER) to create the WS-RM service.

In the consumer system, use the URL of the provider web service to create the consumer proxy. Use SOA Management (transaction SOAMANAGER) to create a logical port. A program can use the consumer service to send a message to the provider service.

In summary, the steps are:

**In the Provider System:**
1. Create a service interface in ES Repository.
2. Generate the ABAP proxy based on the service interface.
3. Add code to the ABAP proxy.  
   In this case, we create a simple program to insert data into a table.
4. Use transaction SOAMANAGER to create a Web service endpoint for the ABAP proxy.
5. Use the Web Services Navigator to test the Web service.

**In the Consumer System:**
1. Create a consumer proxy using the URL of the provider Web service.
2. Use transaction SOAMANAGER to create a logical port for the consumer service.
3. Create a program to call the consumer service.

### 4.1 Create the Provider Application

1. In ES Repository create a service interface for the provider system. Create and activate the following service interface:

![Service Interface](image-url)
This interface is based on the message type:

![Message Type Interface](image1.png)

2. Go to transaction SPROXY to generate an ABAP proxy based on the service interface:

![Object Navigator](image2.png)

Save and activate.
3. Provide code to the ABAP proxy.
   In this case, create a simple program to insert data into a table.
   a. Open the proxy and double-click on the Provider Class:

   ![Image of the ABAP proxy interface showing the Provider Class and method.

   b. Select the Method tab and double-click on the method name:
c. Insert code into the proxy:

![Class Builder: Class ZWS_CL_ORDER_ASYNC_IN Change](image)

```java
// Insert code into the proxy.
```

d. Save and activate the proxy.

4. Use transaction SOAMANAGER to create an endpoint.
   a. On the application client of the provider system, enter transaction SOAMANAGER. A Web Dynpro application starts in a browser.
   b. Select the Business Administration tab and click on Web Service Administration:

![SOA Management](image)

5. Enter the following search criteria:

   Search by: Service
   Search Pattern: Zorder_Async_IN (this value can be obtained from SPROXY)
   Field: Internal Name

   The following result is displayed:
d. Select the proxy name and click on *Apply Selection*:

![Select the proxy name and click on Apply Selection](image1)

```
Select the proxy name and click on Apply Selection:
```

e. Scroll down, select the *Configurations* tab and click on *Create Service*:

![Scroll down, select the Configurations tab and click on Create Service](image2)

```
Scroll down, select the Configurations tab and click on Create Service:
```

f. Enter a service name and a binding name:

![Enter a service name and a binding name](image3)

```
Enter a service name and a binding name:
```
g. Scroll down, select **User ID/Password**, and **Save**:

![Web Service Configuration of Service Definition: ZORDER_ASYNC_IN](image)

h. The new service is displayed:

![Search Results: ZORDER_ASYNC_IN](image)

The Web service for the provider proxy is now ready for use.
i. To obtain the URL for the web service, select the tab **Overview** and click on **Open WSDL document...**:

The following URL is displayed in the browser:

```xml
<?xml version="1.0" encoding="utf-8" ?>
<wso:definitions targetNamespace="http://test.com/ws/abap"
  xmlns:wso="http://schemas.xmlsoap.org/wsdl/
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tns="http://test.com/ws/abap">
  ...
</wso:definitions>
```

Tip

You can copy this URL and paste it to a notepad. This URL is necessary to create the consumer proxy. Use this URL to test the Web service in the following step.

5. Use the WSNavigator to test the Web service.

You can use any WS-RM enabled client tool, for example WSNavigator available in a SAP NetWeaver 7.1, EHP 1 system or above.

Note

WSNavigator in SAP NetWeaver 7.10 or below cannot execute services using WS-RM. This capability is only available as of SAP NetWeaver PI 7.1, EHP 1.

a. In a SAP NetWeaver 7.1, EHP 1 system, execute WSNavigator.
b. Enter the URL obtained from the previous step and click Next:

![Service test interface](image)

**Tip**

Logon is required. Enter a valid user with authorization to execute the Web service on the provider system.

c. Click Next:
d. Enter test values and click Next:

![SAP NetWeaver WS Navigator](image)

- order_MT
- order_no: test
- customer_no: test
- customer_name: test

e. The status Delivering is displayed. Once complete, Delivered is displayed:

![Service test](image)

f. Check the successful execution by displaying the table content:

![Data Browser: Table ZTEST_ORDER](image)

Check the message delivery using transaction SXMB_MONI on the provider system:
4.2 Create the Consumer Application

1. Create a consumer proxy using the URL of the provider Web service.

   Note
   You need the URL saved in previous step of section 4.1.4.i.
   a. Logon to the consumer system, and call transaction SE80.

   Important
   In the consumer system WS-RM must be enabled, too.
   If this is not the case, refer to section 4.1.

   b. Select Local Objects and navigate to Service Consumer.
      Right-click and select Create:
c. Select **URL/HTTP Destination** and click **Continue**:

![Select source diagram]

d. Paste the URL obtain from the provider Web service, and click **Continue**:

![Select URL diagram]
e. Enter a prefix for the proxy object and click *Continue*:

![Image of the Enter Package/Request window](image)

f. Click *Finish*:

![Image of the Complete window](image)

**Important**

Enter a user name and a password to access the provider system. This is necessary to retrieve the metadata information of the Web service.
g. The consumer proxy is created:

![Consumer Proxy Creation](image)

Save and activate the consumer proxy.

2. Use transaction SOAMANAGER to create the logical for the proxy.
   a. On the consumer system call transaction SOAMANAGER.
   b. Go to the Application and Scenario Communication tab and click on Single Service Administration:

   **Note**

   The tab and function name is different depending on the release level. However, the names should be similar. The position of the tab and location of the functions are the same. For example, in SAP NetWeaver PI 7.1, the tab is called Business Administration and the function name is Web Service Administration.
c. In the Web Service Administration screen, select:
   - Search by: Consumer Proxy
   - Search Pattern: order_Async_IN (this value can be obtained from the previous step when the consumer proxy is created)
   - Field: External Name

To search click Go.

The following result is displayed:

![Web Service Administration Screen](image)

```
d. Create a logical port for the consumer service:
   i. Select the consumer proxy
   ii. Click on Apply Selection
   iii. Scroll down and go to the Configuration tab.
   iv. Click on Create Logical Port.
```

![Create Logical Port Screen](image)
e. Enter all required fields and click on Apply Settings:
the URL for WSDL Access is the provider Web service URL from step 4.1.4.i.

**Important**
Depending on the SAP NetWeaver release level, you may be asked to enter the binding name also. If this is requested, enter any binding name, for example create_order_binding.

f. Scroll down and enter the authorized user to execute the provider service and save the configuration:
g. The logical port is displayed:

![Web Service Administration](image)

3. Create a program to call the consumer service.
   Find an example of an ABAP program below, created using SE38:

```abap
REPORT ZWS_CREATEORDER_PI1.

DATA: ZCREATEORDER TYPE REF TO ZWS_CO_ORDER_ASYNC_IN,
    m_seq_prot TYPE ref to IF_WSPROTOCOL_SEQUENCE,
    m_seq TYPE ref to IF_WS_CLIENT_SEQUENCE,
    l_wsprot type ref to if_wsprotocol,
    lv_seq TYPE SRT_SEQ_ID.
```
data: INPUT type ZWS_ORDER_MT.

parameters: Order_No(10) type c LOWER CASE.
parameters: Cust_No(10) type c LOWER CASE.
parameters: Cust_Nm(10) type c LOWER CASE.

input-ORDER_MT-ORDER_NO = Order_No.
input-ORDER_MT-CUSTOMER_NO = Cust_No.
input-ORDER_MT-CUSTOMER_NAME = Cust_nm.

try.
  CREATE OBJECT ZCREATEORDER
  EXPORTING
    logical_port_name = 'CREATE_ORDER'.

  * generation of the sequence protocol and the sequence
  m_seq_prot = ZCREATEORDER->get_protocol( if_wsprotocol=>sequence ).
  m_seq = m_seq_prot->create_persistent_sequence( ).

  * start sequencing and get id
  m_seq->begin( ).
  m_seq_prot->set_client_sequence( m_seq ).
  lv_seq = m_seq->get_id( ).
  CALL METHOD ZCREATEORDER->CREATE
    EXPORTING
      INPUT = input.

  * end sequencing and commit work
  m_seq->end( ).
  cl_soap_tx_factory->commit_work( ).
  write:/ 'Order Successfully Sent'.

CATCH CX_AI_SYSTEM_FAULT
  CX_AI_APPLICATION_FAULT.
  write : 'Error during proxy call'.
  exit.
ENDTRY.

Note the parts of the program marked in red.

- **ZWS_CO_ORDER_ASYNC_IN**: the proxy name as it was created in SE80 above.
- **ZWS_ORDER_MT**: the ABAP structure containing the input data. This is automatically created during the proxy creation process.
- **CREATE_ORDER**: logical port name created in SOAMANAGER in a previous step.
- **CREATE**: the operation name when the service interface is created in the ES Repository. You can also see this name in the ABAP proxy as the method of the class.
- **commit_work( )**: This is necessary for asynchronous message processing.
4. Execute the consumer program:
   a. In a SAP NetWeaver 7.1, EHP 1 consumer system:

   ![Consumer Program Execution](image.png)

   Result in the provider system:

   **Data Browser: Table ZTEST_ORDER**

   ![Table Display](image2.png)
b. In a SAP NetWeaver 7.00, SP 18 consumer system:

Result in the provider system:

4.3 Monitoring and Debugging

WS-RM has its own runtime time engine. Consequently, monitoring and debugging includes a set of different tools in addition to the traditional ABAP and Integration Engine tools.

In this section, we describe some of the available tools.

In order to determine the tools, an overview of the message flow during WS-RM runtime can be helpful. Although the diagrams below outline the flow in SAP NetWeaver PI Integration Server, it is also applicable to the Application Servers since the Integration Engine is also part of the ABAP stack.

Sending message from the network to WS-RM:
Network → ICM → WS Adapter → bgRFC → WS Adapter → Central Pipeline

Sending message from WS-RM to the network:

Central Pipeline → WS Adapter → bgRFC → Network (via HTTP Post)
To monitor the messages, we can use the message processing paths indicated above to determine the tools to use.

1. ICM: monitoring the ICM for incoming messages.
   ICM can be monitored in AS ABAP or AS Java:
   - ABAP: transaction SMICM
     In the main menu choose Goto → Trace File → Display...
     ![Trace File Screenshot]
     Change the trace level by using:
     Goto → Trace Level → ...
     ![Trace Level Screenshot]
- Java: http://<server>:5xx13 (Management Console)

**Note**

Use the following information to create administrator users with authorization to view ICM from the browser:

*Creating Administration Users*
2. Transaction SXMB_MONI or SXI_MONITOR: This is the existing tool used to monitor Integration Engine messages. In this monitor you can display payload information.

![Monitor for Processed XML Messages](image)

3. Call transaction SXMB_MONI or SXI_MONITOR, scroll to the right to display the columns **Queue Sequence** and **Queue ID**. To get additional monitoring information, click on the value of these:

- WS-RM messages are sent as sequences. Multiple messages are sent in specific sequences with guaranteed delivery. SOAMANGER’s sequence monitor provides information about errors during message processing. You can restart or terminate messages in this monitor.
- WS-RM is processed by using background RFC. Monitoring the bgRFC is important to get additional details. The logging of successfully processed asynchronous WS-RM messages is not stored, which is why for successful messages, we do not see anything in the monitor.
Note

Select transaction SBGRFCMON to monitor bgRFCs.

These two monitors give you additional information of the processed messages. When an error occurs, the error details are displayed.
4. **Transaction SRT_TOOLS:** This tool provides a centralized location for access to SOA runtime tools.

### SOA Runtime Tools

<table>
<thead>
<tr>
<th>Monitoring</th>
<th>WS Configuration</th>
<th>Technical Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitor for Processed XML Messages</td>
<td>Display Web Service Runtime Configurations in Current Client</td>
<td>Check Administration Environment of SOAP Runtime</td>
</tr>
<tr>
<td>Sequence Monitor</td>
<td>Pending SOAP Runtime Configuration Tasks</td>
<td>Technical Configuration of SOAP Runtime</td>
</tr>
<tr>
<td>bgRFC Monitor</td>
<td>Extended Display of Service Configurations in Current Client</td>
<td>bgRFC Configuration</td>
</tr>
<tr>
<td>Display Web Service Logging and Tracing</td>
<td>SOA Manager</td>
<td></td>
</tr>
<tr>
<td>SRTUTIL Administration</td>
<td>Integration Directory Cache</td>
<td></td>
</tr>
</tbody>
</table>

For monitoring:

a. Monitoring for Processed XML Messages: This is the same as the monitoring tool which can be accessed using SXMB_MON1.

b. Sequence Monitor: This is the same tool as discussed above.

c. bgRFC Monitor: This is the same tool as discussed above.

d. Display Web Service Logging and Tracing: This tool monitors the messages and trace information as defined by the configurations in *Technical Configuration of SOAP Runtime*. The logging and trace level are configured there.

e. SRTUTIL: Display messages based on users.
5. Transaction SOAMANAGER → Monitoring: The SOAMANAGER tool is a Web Dynpro application which provides not only the administration tools of managing consumer and provider services, as seen in the previous sections, but it also provides monitoring, logs and traces.

- Message Monitoring: This is the equivalent to SXMB_MONI.
- Sequence Monitoring: This is also similar to the sequence monitoring in ABAP, as mentioned above. However, it provides a little different user interface.

6. Transaction SOAMANAGER → Logs and Traces:

This provides detailed message information based on the configuration in Logs Configuration and Trace Configuration.
Example:

7. ABAP Runtime Error: It might also be necessary to go to the runtime error log to determine the problem.

ABAP Transaction: ST22
4.4 Use WSNavigator to Debug Web Services

Use WSNavigator to Debug Web Services:

1. Change trace level to Debug on for the packages:
   - com.sap.engine.services.webservices
   - com.sap.engine.services.wsrm

2. Execute the service using WSNavigator.
3. Go to the default trace in the Log Viewer:
   - a. Enter com.sap.engine.services in the filter for column Location:
b. Using SOAMANAGER:
   
   i. In /nwa:

   ![SOAMANAGER Interface](image)

   ii. In ABAP:
       
       TX:  ST22
       
       To examine the dumps (if any) generated by the Web Service call.

       TX: SE38 - report SRT_ADMIN_CHECK
       
       To determine any missing Web Services configurations.

   c. Where to monitor?

   **Sender System:**
   SXMB_MONI (aka SXI_MONITOR)
   
   Use the default display:

   ![Monitoring Display](image)

   This display provides the Queue/Sequence id and the queue name/ID:
d. By clicking on the sequence ID, you open the Sequence Monitor, to get additional information about the message. You can also examine errors and start/end the sequence.

By clicking on the sequence ID, you open the Sequence Monitor, to get additional information about the message. You can also examine errors and start/end the sequence.

Sequence Monitor: Sequence Overview

![Sequence Monitor](image)

By clicking on the Queue ID, you can examine the bgRFC queue where the message is processed. The bgRFC monitor indicates if our message is waiting in the queue, and the reason it might be stuck.

Monitor for bgRFC Units

![Monitor for bgRFC Units](image)

In addition to monitoring the message in the AS ABAP, you can also monitor the message in transaction SOAMANAGER in the web browser.

In SOAMANAGER, select Monitoring → Sequence Monitoring:

![SOA Management](image)

(Message Monitoring is a link to the ABAP’s SXMB_MONI.)
In SOAMANAGER’s Sequence Monitor, you can copy and paste the Sequence ID from SXMB_MONI, or any other search criteria:

Here, you can delete, close or restart the sequence. You can also examine additional detail about the sequence:

Receiver System:
Similarly to the sender System, SXMB_MONI and SOAMANAGER can be used to monitor the messages.
www.sdn.sap.com/irj/sdn/howtoguides