How To Configure SAML Authentication for SAP NetWeaver Process Integration 7.1

Applicable Releases:
SAP NetWeaver Process Integration 7.1x and higher

Topic Area:
SOA Middleware

Capability:
SOA Management

Version 1.0
April 2009
## Document History

<table>
<thead>
<tr>
<th>Document Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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><em>Example Text</em></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><em>Example text</em></td>
<td>Emphasized words or phrases in body text, graphic titles, and table titles.</td>
</tr>
<tr>
<td><em>Example text</em></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><em>Example text</em></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><em>&lt;Example text&gt;</em></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><strong>EXAMPLE TEXT</strong></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><img src="caution.png" alt="Caution Icon" /></td>
<td>Caution</td>
</tr>
<tr>
<td><img src="note.png" alt="Note or Important Icon" /></td>
<td>Note or Important</td>
</tr>
<tr>
<td><img src="example.png" alt="Example Icon" /></td>
<td>Example</td>
</tr>
<tr>
<td><img src="tip.png" alt="Recommendation or Tip Icon" /></td>
<td>Recommendation or Tip</td>
</tr>
</tbody>
</table>
# Table of Contents

1. Business Scenario............................................................................................................... 1  
   1.1 Introduction................................................................................................................... 1  
   1.2 Scenario Description .................................................................................................... 3  
2. Background Information..................................................................................................... 4  
3. Prerequisites ........................................................................................................................ 4  
   3.1 Software........................................................................................................................ 4  
      3.1.1 Supported releases.......................................................................................... 5  
      3.1.2 ABAP Kernel and Cryptographic library........................................................... 5  
   3.2 Relevant SAP Notes..................................................................................................... 8  
4. Sender-Vouches Subject Confirmation Method for SAML Token Profiles....................... 9  
5. Configuration Steps in Detail ........................................................................................... 11  
   5.1 Setup Web Service Security....................................................................................... 11  
   5.2 Configuring Trust Relationship for ABAP Systems .................................................... 12  
   5.3 Enable Principal Propagation ..................................................................................... 16  
   5.4 Maintain Trusted Issuer ........................................................................................... 17  
   5.5 Configure Communication.......................................................................................... 21  
6. Monitoring .......................................................................................................................... 34  
7. Appendix ............................................................................................................................ 37  
   7.1 Troubleshooting – Error Scenarios............................................................................. 37  
      7.1.1 Problems and Solutions................................................................................. 37  
   7.2 Additional Information................................................................................................. 44  
      7.2.1 Help Documentation....................................................................................... 44  
      7.2.2 Upcoming Features......................................................................................... 44  
   7.2.3 Tips ..................................................................................................................... 45
1. Business Scenario

1.1 Introduction

Entering usernames and passwords on the Internet has become a significant security challenge. As single sign-on provides access to many resources once the user is initially authenticated ("keys to the castle"), it increases the negative impact in case the credentials are available to other persons and misused. Therefore, single sign-on requires an increased focus on the protection of the user credentials, and should ideally be combined with strong authentication methods, e.g. smart cards, Kerberos, SSL etc. It also, reduces sharing and impersonation of usernames and passwords, eliminates usage barriers and risky proprietary SSO implementations that are vulnerable to attacks.

Historically, the majority of Web services have been based on the same enabling technology that underlies the Web, namely HTTP (Hyper Text Transfer Protocol). As a result, common technologies that secure Web applications, such as Basic Authentication and SSL (Secure Sockets Layer), work equally well with web services. These security technologies have been effective for years for all sorts of online business transactions, and they work equally well for Web services.

Security at the transport layer: SSL, however, does have limitations. SSL for Web services is an all-or-nothing proposition: it often secures the entire wire protocol rather than just the SOAP (Simple Object Access Protocol) message sent over the protocol. It doesn't let developers apply different levels of security to different parts of a document. Because of its point-to-point structure, SSL doesn't support chained services or workflow applications where user credentials can be passed through each stop in a transaction chain. This leaves the messages unsecured at each intermediary checkpoint. SSL also does not support the concept of an audit trail. There are different ways you can use standard HTTP authentication mechanisms, such as

- Basic Authentication (User ID and Password)
- Http over SSL
- X.509 Certificates
- SAP Assertion Tickets (Logon Tickets)

Security at the message layer: The answer to providing both message- and transport-level security lies with the OASIS (Organization for the Advancement of Structured Information Standards) standard, Web Service Security, released as a full industry-recognized recommendation in April 2004. WS-Security defines a mechanism for adding transport independence and different levels of security to SOAP messages. There are many SOAP message authentication mechanisms, such as

- Username Token
- X.509 certificate Token
- XML Signature
- XML Encryption
- SAML 1.1 Token
- SAML 1.1 Assertion issued by a central Security Token Service

If there is a need to apply message integrity and confidentiality at a fine-grained level instead of applying to the entire SOAP message, XML signature and encryption can be used to protect the SOAP body, header block, or portions of either. If there is a need to use different forms of authentication, then message-level security authentication tokens can be used, such as username token, X.509 token, SAML token or SAML Holder of the Key (STS).
**WS-Security Scenario**

Web Service Security describes mechanisms to provide message integrity and confidentiality for SOAP messaging. It leverages existing standards like XML Signature, XML Encryption to cover a wide variety of security models and encryption technologies. It plugs into the SOAP architecture and extends the SOAP header information in the protocol with additional security information such as encryption keys, digital signatures and authentication information. WS-Security addresses end-to-end security through the definition of the security header format for SOAP messages that is designated by the opening and closing `<Security>` block. The specification builds upon existing and approved approaches to distributed systems security such as public key encryption, digital signatures or certificates. For protecting messages, X.509, SAML 1.1 and Secure Conversation are supported.

WS-Security does not invent any new security mechanisms – it simply defines how existing ones, such as XML Encryption or XML Signature, should be applied in the context of a SOAP message. Figure 1 illustrates a common usage scenario for WS-Security: The SOAP message is signed with the private key of sender to ensure integrity of the content. Based on the signature, the recipient can verify the identity of the sender and that the data has not been modified by an intermediary during transit. More precisely, the sender has signed two elements of the SOAP message: The SOAP `<Body>` with the application data and the `<Timestamp>` element of the message, which contains the message creation date and time to let the recipient decide whether the message is ‘fresh enough’ to be processed or not.

![Figure 1: Signed SOAP Message with WS-Security](image)

Based on the certificate (like X.509, SAML…) contained in the `<BinarySecurityToken>` element that belongs to the private key used to sign the message, the recipient can determine the sender’s identity and decide whether to trust it or not. Successful verification of the sender’s identity requires a pre-established trust relationship between both parties. Sender and recipient usually trust a common Certification Authority (CA) and trust the certificates issued by that CA.

This document explicates a way of authenticating web services using SAML 1.0 token mechanism.

SAP Netweaver Process Integration 7.1 contains an interoperable security solution to allow systems integration with great ease and minimal resources using Security Assertion Markup Language (SAML). The SAML is an XML-based framework for exchanging security information. This security information is expressed in the form of assertions (security related information) about subjects (Business users), either human or computer, that have identities in some security domain. One example of a subject is a person, who is identified by his or her email address in a particular Internet...
DNS domain. Assertions convey information about authentication acts performed by subjects, attributes of subjects, as well as authorization decisions about whether subjects are allowed to access certain resources.

SAML provides a flexible and extensible framework for business and Web services to exchange security information, such as authentication, authorization, and attributes, about their users. For example, it allows one service to vouch for a user's authentication to another service. This provides useful activities, such as single sign-on to multiple services.

SAML lets one service inform another service whether a user is permitted to access a given resource. This allows policy enforcement to be decoupled from resource services. It also provides attribute information about a user, such as work-site address, to be communicated between services as the user moves from one site to another.

Within a single company SAP Logon Tickets may be used for Single-Sign-On (SSO). But if SSO between different companies is needed (authentication over „Domain Boundaries“) the use of SAP Logon Tickets is not practical any more. SAML addresses the problem of authentication and authorization across „Domain Boundaries“.

Web service infrastructure in Process Integration 7.1 supports both point-to-point and mediated (over PI) scenarios. It enables to use the sender-vouches subject confirmation method to confirm a subject with SAML token profile authentication and Web service Reliable messaging (WS-RM) for Principal Propagation.

For this subject confirmation method, the SAP NetWeaver PI (Web Service intermediary system) acts as a SAML assertion issuer. SAP NetWeaver PI system authenticates the Web Service (WS) client and forwards to the back-end WS provider the authentication information for the WS consumer using SAML Token profile. The WS provider, in turn, authenticates access based on its trust relationship with the intermediary system.

You would like to use this feature in your SAP Netweaver PI 7.1 system.

⚠️ Note

SAP NetWeaver PI 7.1 system is referred to as “PI system”, WS provider system as “Provider” and WS consumer system as “Consumer” in the remainder of this document.

### 1.2 Scenario Description

To better understand the whole the process flow it is necessary to take a simple business scenario where Consumer sends a customer number synchronously to the Provider expecting the customer details as a response. PI system invokes a service on the Provider to get the details. The main purpose of the scenario is to show how the user identities can be propagated from the Consumer to the Provider via the intermediary server, Process Integration.

Consumer triggers the process by sending the customer number via ABAP client proxy exposed as a web service using WS runtime. At Provider system a server ABAP proxy exposed as a service handles the client request and responds with the details of the customer.

The following graphic describes the important application components and their communication with the PI system for a typical three system landscape.
2. Background Information

This guide explains how to configure SAML authentication using SAP NetWeaver Process Integration 7.1. In this document we will use SAML 1.0 Sender Vouches Assertion method to configure web service security between AS ABAP systems. Where information is available otherwise, the guide will refer to the same rather than to repeat it.

3. Prerequisites

To read the specification of SAML, see OASIS documentation at

To get a basic understanding of what SAML is all about, see SOAP Message Level Authentication

This guide assumes that the Consumer and Provider ABAP Proxy scenario is configured beforehand using the “Help Documentation” given in the “Appendix” section.

3.1 Software

This section provides the details of supported releases for the applications (Consumer, Integration Server and Provider) and the version details of ABAP service pack, ABAP Kernel and Crypto library. The technology stack of backends can be AS ABAP, AS Java, or external system.

This guide makes the following assumptions:

- An ABAP back-end is used at the consumer and provider ends
- SAP NetWeaver PI 7.1 is installed.
3.1.1  Supported releases
Consumer    – AS ABAP 7.0 >= SP14
Integration Server  – AS ABAP 7.1 and higher
Provider    – AS ABAP 7.0 >= SP14

3.1.2  ABAP Kernel and Cryptographic library
ABAP Kernel 700.REL   – Patch level 179 is minimum
ABAP Kernel 710.REL   – Patch level 116 is minimum

Note
Download the latest Kernel patch from SMP (Service Market Place)

SAP Cryptographic Library  – 1.555.24 or higher must be installed on all systems involved.
You can check this in transaction STRUST, menu Environment->Display SSF Version.
For example:
SSFLIB Version 1.555.24; SECUDEm SAPCRYPTOLIB –
SNC for SAP Server components and SSL –
Version5.5.5C (c) SECUDEm GmbH 1990-2004
For more information about the installation of SAP Cryptographic Library, see
 Installing the SAP Cryptographic Library on the SAP Web Dispatcher  and check SAP Note 662340.

After installing the cryptographic library, create the PSEs (Personal Security Environment) in Trust-
Manager by using the transaction STRUST. This step ensures that the cryptographic keys and
certificates required for the application are stored in the PSE and this step needs to be done on all
involved systems (Consumer, Integration Server and Provider).
Make sure you check the system PSE health, before installing crypto library, as given below:

Before installation of Cryptographic library
1.  Call transaction STRUST as shown below. You should see the green node by expanding
System PSE. This ensures that the system PSE is stored in the database and distributed to
application servers.
2. Double click on Owner (CN=XID, for example as shown in Figure 3). This will show the system certificate details like Owner, Issuer, Serial Number etc. as shown below:

![Figure 3: Trust Manager](image)

- Owner: [Details]
- Issue: [Details]
- Serial Number: [Details]
- Valid From: [Details]
- Valid To: [Details]

Add to Certificate List
After installation of Cryptographic library

3. After installing the crypto library, call transaction STRUST and create all PSEs (SNC, SSL server, SSL client, SSF application and WS Security) which are displayed with Red Cross mark as shown below:
How To Configure SAML Authentication for SAP NetWeaver Process Integration 7.1

Figure 5: Maintain PSE

For more information, see Creating PSEs and Maintaining the PSE Infrastructure.

3.2 Relevant SAP Notes

1170238 – Central note for WS Security on 7.10
1176558 – Central note for WS Security on 7.00
1280825 – WS adapter: Dump STRING_LENGTH_TOO_LARGE
1175833 – Transaction SECSTORE displays many SOAP runtime entries
1108662 – Missing authorization for SYST or RFCPING
1260080 – Activating PI objects leads to no change in ABAP
1247500 – bgRFC scheduler terminates w/ arithmetic overflow exception
662340 – SSF Encryption Using the SAPCryptolib
1178531 – SYSTEM_CORE_DUMPED, exception CX_SY_MESSAGE_IN_PLUGIN_MODE
4. Sender-Vouches Subject Confirmation Method for SAML Token Profiles

The SAML Token Profile authentication enables you to confirm a subject in SAML with the sender-vouches subject confirmation method. This method enables SSO for Web services by using a SAML assertion to forward authentication information acquired in an initial logon. You can use the sender-vouches confirmation method for SSO scenarios where the WS intermediary system has a trust relationship with the back-end system. This scenario defines four different entities: (1) a client, (2) an intermediary, (3) SAML issuer, and (4) a back-end system that is the WS provider.

For an overview of the system interaction for this scenario, see the figure below:

**Figure 6: Sender-Vouches Subject Confirmation Method**

The following steps describe in more detail the lifetime of a request using the SAML sender-vouches profile.

1. The client sends a request to the intermediary. This request can be of any kind but must contain valid authentication information to log the client on to the intermediary.

2. The intermediary authenticates the client. To process the request, the intermediary needs to retrieve information from the back-end system using Web Services forwarding mechanisms for the client’s authentication information.

3. To forward the client’s authentication, the intermediary needs to add a SAML assertion to the request. This assertion is provided by the issuer. To get it the intermediary needs to forward all necessary login information to the issuer, which in return creates the SAML assertion.

4. The assertion is added to the Web service request. To vouch for the integrity of the SAML assertion and the payload of the Web service request both are signed by the intermediary using a digital signature. The intermediary is able to vouch for the SAML assertion because there is an explicit trust relationship between the back-end system and the intermediary, which enables the back-end system to verify the digital signature.

5. The Web service request containing the SAML assertion is now sent to the back-end system.

6. The back-end system attempts to verify the SAML assertion. Other than checking the correctness of the SAML assertion, the back-end system also verifies that the issuer is trusted and there is an existing trust relationship between the intermediary and the back-end system. After successful verification, the client is logged on to the system and the request is processed.

7. The back-end system sends a response to the intermediary. The intermediary uses the received data to complete the client’s request and send a response to the client.
The following excerpt shows what a security header including a SAML assertion using the sender-vouches subject confirmation method looks like.

```
<wsse:Security>
  <saml:Assertion AssertionID="SAML_ID" Issuer="www.example.org" ...>
    <saml:Conditions NotBefore="..." NotOnOrAfter="..."/>
    <saml:AuthenticationStatement
        AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:unspecified"
        AuthenticationInstant="2009-03-19T...Z"
        >
      <saml:Subject>
        <saml:NameIdentifier NameQualifier=""
            Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
            Your Business User that is propagated</saml:NameIdentifier>
      </saml:Subject>
      <saml:SubjectConfirmation>
        <saml:ConfirmationMethod>
          urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
        </saml:ConfirmationMethod>
      </saml:SubjectConfirmation>
    </saml:AuthenticationStatement>
  </saml:Assertion>
  <wsse:SecurityTokenReference wsu:Id="STR1" ...> ...
</wsse:SecurityTokenReference...>
  <wsse:BinarySecurityToken ...> ... </wsse:BinarySecurityToken>
  <ds:Signature>
    <ds:SignedInfo>
      <ds:Reference URI="#STR1"> ... </ds:Reference>
      <ds:Reference URI="#body"> ... </ds:Reference>
    </ds:SignedInfo>
  </ds:Signature>
</wsse:Security>
```
5. Configuration Steps in Detail

This section covers the configuration steps required in back-end systems and PI system for message processing with Integration Server communication.

5.1 Setup Web Service Security

To create the configuration for WS Security, execute the report “WSS_SETUP” by calling the transaction SA38 in all the systems (Consumer, Integration Server and Provider) involved, as shown below:

![ABAP: Program Execution](image)

Figure 7: WS Security Configuration

The successful result of the report is shown below:

![Program WSS_SETUP](image)

Figure 8: WSS_SETUP result

The report creates a user, 'DELAY_LOGON', which does not have any authorizations on Integration Server. Also it stores the user password in the secure storage. Whenever Consumer of an SAP system logs on to a web service with message based authentication such as SAML Token, the Internet Communication Framework (ICF) initially performs the logon using the technical user 'DELAY_LOGON' that is stored in it. A direct logon with the authentication data contained in the SOAP document is not possible, since the ICF cannot access SOAP data. It requires an HTTP authentication (Transport level) and therefore it uses the DELAY_LOGON user. All ICF nodes, created by SOAP runtime with message authentication, use DELAY_LOGON user for initial authentication. After this step the SOAP runtime performs a user switch and logs on the user specified in SAML Token as shown below:
CAUTION

Never change the password of DELAY_LOGON user on Integration Server and do not assign any roles or profiles to this user.

5.2 Configuring Trust Relationship for ABAP Systems

Each communication step along the way from the sender to the receiver requires a separate authentication for each messaging component before the message is executed. This implies that the message is executed under the same user in all participating messaging components. Wherever SAML assertion is used for authentication between a sending and a receiving messaging component, a trust relationship, between the underlying application servers first, needs to be configured.

As a prerequisite, the X.509 client certificates for the WS Security PSEs must have been signed by a Certification Authority (CA), so that encryption can be used. The certificate contains the CA v3 extension Subject Key Identifier.

Since a distinction is made between the sender (client) and the receiver (server) side, two configuration variants apply:

- AS ABAP client configuration
- AS ABAP server configuration

Note

Different components (Consumer, Integration Server and Provider) act either as client [C] or server [S] depending on the connection, as shown below:

Request:  AS ABAP WS consumer [C] → [S] Integration Server [C] → [S] AS ABAP
          
Response: AS ABAP WS provider [C] → [S] Integration Server [C] → [S] AS ABAP
**Consumer**

**Procedure**

1. On ABAP Client [C], call transaction STRUST and export the certificate as shown below:

![Figure 10: Export WS Consumer system certificate](image)

2. Choose ‘Binary’ file format and save the certificate to local desktop as shown below:
You have made the certificate available as a file, which you can later import into the ABAP Server [S] system.
3. On ABAP Server [S], call transaction STRUSTSSO2 and import the certificate as shown below:

![Figure 12: Import certificate into ABAP Server [S]](image_url)

4. Select ‘Binary’ file format and import the client certificate which is saved as file from step 2

5. Click on ‘Add to Certificate List’ button to add this certificate to the list

6. Click on ‘Add to ACL’ button to add the client system to Access Control List of server as shown below:
Save the data now and as a result, you will be able to see the ABAP Client [C] system added as an entry to the Access Control List window as given below:

![Access Control List window](image)

**Figure 14: Access Control List window**

You need to perform the above steps for every client-server combination as explained in the above important note for establishing the SSO trust between all systems.

### 5.3 Enable Principal Propagation

Call transaction SXMB_ADM (Integration Engine - Administration) and run ‘Configure Principal Propagation’ to activate principal propagation.

- This will create a service user XIPPU$ER with role SAP_XI_APPL_SERV_USER assigned to the same and a RFC destination SAPXIP$<Client No>.

- Activation has the effect that the user context in message processing remains the same for all asynchronous locks and restarts. The context of the user to be propagated is impersonated again after a message restart or during batch processing only if principal propagation is enabled for the involved messaging component (The ABAP messaging components are the Integration Server and the Integration Engines of the sender and receiver systems executing ABAP proxies).

This step needs to be done on Consumer, Integration Server and Provider systems as shown below:
5.4 Maintain Trusted Issuer

Determine the issuer

1. In the Consumer, run report ‘WSS_INFO’ using transaction SA38 to determine the data that the accepting system requires. This provides information about how the issued SAML assertion looks like, as shown below:

```
Typ der externen ID: SA
Prefix des externen Namens: EC6/001:
Ext. Name des Ausstellers: DN=EC6

SAML Aussteller: EC6/001[□]
SAML name qualifier:
PSE im lokalen System: SYSPBE
SAML Signaturezertifikat:

----BEGIN CERTIFICATE-----
MIIECDCSAgicBgAsCAIwDwIBADANBgkqhkiG9w0BAQEFAAOCAAAoI:
MDc9NTAAMRAwDgYDVQQKEwNAdW50ZW4gV2VjdXJlczUgRGlnaHRz
MBo9MRAwDgYDVQQKEwNLg==

----END CERTIFICATE-----
```
The above procedure determines the issuer of SAML assertion and the certificate used to sign the SAML assertion. For instance, the issuer is EC6 client 001 and certificate used is enclosed in the tags ‘---BEGIN CERTIFICATE---’ and ‘---END CERTIFICATE---’ from the Figure 14. This information will be used for setting up the Integration Server to accept the SAML token profiles.

**Preparing the Integration Server to accept SAML Token Profiles**

2. For all propagated users, a user mapping has to be maintained. User mapping clarifies following questions:
   - Do I trust the issuer of the assertion, and hence the user that is the subject of the assertion?
   - Which user should be logged in?

Optional, maintain user mapping either directly in table USREXTID, or via table view VUSREXTID. In the Integration Server, call transaction SA38 and execute the report RSUSREXTID to maintain a user mapping for the business user to be propagated as shown below:
How To Configure SAML Authentication for SAP NetWeaver Process Integration 7.1

In the next screen, enter the following values to maintain user mapping.

Figure 17: User Mapping
How To Configure SAML Authentication for SAP NetWeaver Process Integration 7.1

**Tip**

Clear or remove the default values of all the fields before entering the following values.

- **User**: Business user that needs to be propagated
- **External ID type**: SA
- **Prefix of External Name**: `<SID>/<Client>::` (Copy the value of ‘Prefix of External Name’ from step 1 screen above and paste it in this field)
- **Optional Issuer’s Name**: Copy the value of the ‘External Name Of Issuer’ from Step 1 screen above and paste it in this field

Uncheck the fields ‘Only Users without Ext. Names’ and ‘Test Mode’.

Click on EXECUTE and verify the results on the next screen.

3. Optionally, call transaction SM30 and enter the Table/View name as VUSREXTID to display and verify the user mapping for the business user. In the next pop-up window select ‘External ID type’ as ‘SA’.

You will see your business user present in the table with the status activated as shown below:

![Display View "Assignment of External ID to Users": Overview](image)

**Figure 18: Display User Mapping**

4. Go to transaction SA38, and execute the report WSS_INFO. This information will be used for setting up the Provider to accept SAML token profiles as shown below:
Preparing the Provider to accept SAML Token Profiles

5. Go to transaction SA38 and execute the report RSUSREXTID to maintain user mapping for the business user. Repeat the steps 2 and 3 on Provider also.

5.5 Configure Communication

Follow the steps below to configure the ABAP backend as a WS Provider assuming that the proxy scenario is already configured using the “Help Documentation” given in the “Appendix” section.

Use SOAMANAGER to configure the endpoints for the Provider proxy

1) Logon to the Provider, go to transaction SOAMANAGER. Navigate to Business Administration ➔ Web Service Administration, as shown below:
How To Configure SAML Authentication for SAP NetWeaver Process Integration 7.1

2) The following web page will be displayed in the web browser. Select the Provider proxy service for which you need to create the Logical port and Binding information.

1. Select Search by: Service
2. Enter Search Pattern: This name is from the SPROXY display of the “External Key” of the provider interface.
3. Select Field: External Name or Both Names.
4. Click on button “Go”.
5. With the Search Result, select the service
6. Click on button “Apply Selection”.

This procedure is shown below:

3) On the lower half of the screen, go to Configuration Tab → Create Service.
4) Enter any service and binding names for the web service:

![Figure 22: Create Service](image)

5) On the lower half of the screen,
   a) Select “SAML 1.1 Sender Vouches Assertion
   b) Save the configuration

![Figure 23: Creating Binding Information](image)
How To Configure SAML Authentication for SAP NetWeaver Process Integration 7.1

6) Go to Overview tab and click on “Open WSDL document for selected binding”. This will open a browser window with the WSDL document where you can copy the URL of the web service and paste it to temporary location. This URL is needed while configuring the receiver communication channel in Integration Directory.

---

**Note**

In the case of “Secure Conversation”, you need to select the transport guarantee type as “Secure Conversation” instead of “No Transport Guarantee”.

---

Figure 24: Message Authentication
Figure 25: Web Service URL

Example


Host : inst50053318.blr4.sap.corp
Port : 8000
URL Access Path:

/sap/bc/srt/wsdl/bndg_DE07133F1607BBF1B1B600143850A9D0/wsdl11/allinone/ws_policy/document?sap-client=001

Configure SAML Authentication in Integration Directory

Follow the below steps in Integration Directory assuming that the configuration scenario is already created. In this section you will see how the sender and receiver communication channels can be configured.

7. **WS Sender Communication Channel:** (Consumer)

   Enter the system related information of the PI server. Change the Authentication Method to “SAML 1.1 Sender Vouches Assertion (Authentication in message)” as shown below:
Figure 26: Sender Communication Channel (WS adapter)

8. **WS Receiver Communication Channel**: (Provider)

Enter the system related information of the Provider. Change the Authentication Method to “SAML 1.1 Sender Vouches Assertion (Authentication in message)”. The URL access path can be obtained from the step 6 above.
9. **Sender Agreement**: Create a sender agreement for the sender communication channel, configured in step 7.
Obtain the WSDL URL for the Sender Agreement. This URL will be used in the configuration of consumer proxy.

Menu: Sender Agreement → Display WSDL
How To Configure SAML Authentication for SAP NetWeaver Process Integration 7.1

Figure 29: Display WSDL

Copy and Paste the URL to a temporary location:

Figure 30: WSDL URL for consumer
Example


10. **Receiver Agreement**: Create a Receiver Agreement for the receiver communication channel, configured in the step 8.

![Figure 31: Receiver Agreement](image)

When the PI integration server makes an outbound call, it needs to have the WSDL file. The WSDL file contains interface information as well as WS Policy. WS Policy contains information about the security requirements for the web service (what kind of authentication, use of XML Signature/Encryption). The WSDL is obtained by an HTTP GET request as

1) WSDL access is only granted to authenticated users
2) HTTP GET requests can not be authenticated based on HTTP Headers as no SOAP message is transferred

So a different kind of authentication is needed and this must be a transport authentication. Therefore the Integration Server uses basic authentication with a service user to obtain the WSDL once. During runtime, no basic authentication is used, but SAML is used for authentication. The user should have authorization to access the metadata.
Use transaction SOAMANAGER to configure the ports/bindings for the consumer proxy

Follow the below steps to configure the web service for the consumer proxy:

11. Logon to the Consumer and execute transaction SOAMANAGER. Click on tab “Business Administration” and then “Web Service Administration”.

Figure 32: Web Service Administration

12. Select the consumer proxy:

Figure 33: Search Consumer Proxy

13. On lower half of the page, click on the “Configurations” tab and then “Create Logical Port” button.
14. Create the service and port for the PI system.

Figure 35: Create Service and Logical Port for PI system

Save the configuration:
Figure 36: Save Configuration

When completed, you will see the Logical Port and Service created by you as shown below:

Figure 37: Display Logical Port
For more information on How to configure Consumer and Provider ABAP proxy scenarios, refer to “Help Documentation” in “Appendix” section.

6. Monitoring

Go to transaction SRT_TOOLS for monitoring the processed XML messages. Monitoring the Web service runtime is part of the central monitoring operation. For more information, see Monitoring the Web Service Runtime

You can monitor sequences, processed XML messages (SXMB_MONI transaction), bgRFC units and many more as shown in the below diagram.

![SOA Runtime Tools](image)

Figure 38: SOA Runtime Monitor

Logs and Traces of SOA Manager help in analyzing errors or performance problems. To display logs and traces, see Displaying Logs and Traces

To see the SAML assertion of payload either in Consumer or Provider, you need to set the trace level in SOAMANAGER to ‘Full Trace’. Then you will get the complete message with assertion details.

1. Call Transaction SOAMANAGER in Consumer or Provider systems. This will load a web browser as shown below:
How To Configure SAML Authentication for SAP NetWeaver Process Integration 7.1

2. Go to Logs and Traces tab and click on ‘Trace Configuration’.

3. Set the trace level to ‘Full Trace’ as shown below:

4. Execute your scenario again and go to ‘Logs and Traces’ tab as explained in step 2. Click on Logs/Trace Viewer.
5. Select your message from the displayed list and click on 'Download Request' link to download the complete message. Save the file to local desktop with '.xml' extension.

Open the XML file and the SAML assertion looks like below, for example.
How To Configure SAML Authentication for SAP NetWeaver Process Integration 7.1

7. Appendix

7.1 Troubleshooting – Error Scenarios

7.1.1 Problems and Solutions

7.1.1.1. Execution of report WSS_INFO results an exception “PSE does not exist in database”

**Problem**

When you execute the report WSS_INFO, an exception is thrown saying that “PSE does not exist in database”. The exception is shown below:

```
1.0#X509V3'>MIICUDCCAg8CBgAIABgEIKqQIwDAgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCclgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCglgY1AoAAAABgEBgCg
Figure 45: WSS_INFO Error

Validity
SAP Netweaver PI 7.1 SP06, SAP ECC 6.0 SP 17

Possible Reasons
The cryptographic library is not installed and the system is still using SAPSECULIB which is the default security provider in SAP systems. This library can be used for digital signatures created by the SAP system but not for encrypting data. To have the system encrypt data, you must replace the SAPSECULIB with the SAP Cryptographic Library (SAPCRYPTOLIB).

Solution
Replace the SAPSECULIB with the SAPCRYPTOLIB on the system's application server(s) and set up the system for using encryption. Follow the steps given in SAP Note 662340. After installing the crypto library, call transaction STRUST and create all PSEs (SNC, SSL server, SSL client, SSF application and WS Security) which are displayed with Red Cross mark as given in the Prerequisites section.

7.1.1.2 Execution of Consumer proxy outputs an exception “Message outdated”

Problem
When you execute Consumer proxy by means of a report or a function module the system displays an exception message saying that “CX_WS_SECURITY_FAULT: An exception occurred: Message outdated” as shown below:
**How To Configure SAML Authentication for SAP NetWeaver Process Integration 7.1**

**Validity**
SAP Netweaver PI 7.1 SP06, SAP ECC 6.0 SP 17

**Possible Reasons**
The clocks of the Consumer, PI system and Provider are not synced. The time difference between them is more than 30 seconds. Because of this, older SAML assertion is being sent or WSRM resends the expired message. This assertion is rejected on the Provider side.

**Solution**
Synchronize the clocks of all systems involved and reduce the time skew by less than or equal to 30 seconds.

### 7.1.1.3 Exception “CX_SY_MESSAGE_IN_PLUGIN_MODE: Message E 1S 110 cannot be processed”

**Problem**
While running the scenario, an exception is thrown saying that “CX_SY_MESSAGE_IN_PLUGIN_MODE: Message E 1S 110 cannot be processed in plug-in mode HTTP”. This problem can happen on Consumer or PI system.

**Solution**
Apply the correction attached to SAP Notes 1170238 and 1176558 on Consumer, PI system and Provider systems.
7.1.1.4 System Error while calling Web service adapter: Error when initializing SOAP client application: ‘error_text’

Problem
When you use the Web service adapter (adapter type "WS") to execute a PI call and use the PI integration server for communication ("mediated communication"), you cannot process PI messages. It hangs in the PI message monitor (transaction SXI_MONITOR) with the status "System error". The following error occurs when you display the details for the PI message:

"System error while calling Web service adapter: Error when initializing SOAP client application" as shown below:

```xml
<?xml version="1.0" encoding="utf-8"?>
- <!--
Call Adapter
_- <SAP:Error xmlns:SAP="http://sap.com/xi/XI/Message/30"
xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
SOAP:mustUnderstand="">
  <SAP:Category>XIServer</SAP:Category>
  <SAP:Code area="INTERNAL">WS_ADAPTER_SYS_ERROR</SAP:Code>
  <SAP:P1 />
  <SAP:P2 />
  <SAP:P3 />
  <SAP:P4 />
  <SAP:AdditionalText />
  <SAP:Stack>
    System error while calling Web service adapter: Error when initializing SOAP client application: 'error_text'
  </SAP:Stack>
  <SAP:Retry>M</SAP:Retry>
</SAP:Error>
```

Figure 48: WS Adapter Error

Validity
SAP Netweaver PI 7.1 SP06, SAP ECC 6.0 SP 17

Possible Reasons
There can be many possible reasons as given below:

- The error indicates that the required Consumer configuration does not exist.
How To Configure SAML Authentication for SAP NetWeaver Process Integration 7.1

- There is a configuration error in the PI directory for the relevant receiver agreement. In the PI directory, the fields "User Name" and "User Password" for the receiver agreement in the screen group "Authentication" for the "WS" adapter-specific attributes are not specified or the user entered cannot be used for a login in the target system (he or she does not exist, the password is incorrect, the user is locked, the user is not in its validity period or similar).

Solution

Follow the SAP Notes 1142454 and 1142596. Also refer to the problem 7.1.1.5

7.1.1.5 PI cache refresh problems in connection with WS adapter

Problem

When you execute a Web service (WS) that has been configured using the process integration (PI) framework (more precisely, the PI integration directory (PI directory), the system does not process the message. In the PI message monitor (transaction SXI_MONITOR), messages remain in the "System Error" status (red flag).

The relevant message in the sequence monitor (report SRT_SEQ_MONITOR or transaction SRT_TOOLS, "Sequence Monitor" selection or forward navigation in the PI monitor [as of NetWeaver 7.11]) contains an error message of the following type:

SRT: HTTP-Code 500: ("SRT: Failure in SOAP processing occurred: ("No configuration found")")

OR

SRT: HTTP-Code 404: ("Not Found")

OR

SRT: Initialization of SOAP Runtime failed: "No configuration found"

Validity

SAP Netweaver PI 7.1 SP06, SAP ECC 6.0 SP 17

Possible Reasons

The required Web service (WS) configuration cannot be determined in the involved ABAP systems so that the related PI message remains in the processing queue.

This is an error that occurs during the PI cache refresh processing for objects of the PI adapter type "WS". This error can have different causes, but usually, it occurs due to an incorrectly configured security environment of the interacting systems or due to a problem with the underlying design tie objects (service and proxy definitions). It may also happen if Integration Server faces a problem in getting the WSDL metadata. This process of getting the metadata occurs only once and Integration Server won't try to get the WSDL for subsequent WS calls.

For more information about the reasons and prerequisites, see the related notes.

Solution

Do a dummy change in the Integration Directory -> find the scenario objects (as given below) ->edit ->add a space character in the description field and then after delete it and save ->activate the change list key (as well save it in a notepad) ->On all ends execute the report ->SSRV_CPA_TEST with the new change list key. This report enables you to repeat the PI cache refresh process in the relevant back-end systems. To repeat the PI cache refresh process, you require the PI objects that are
involved in the generation of the WS configuration or the WS configurations. This report then reads these objects from the PI directory and triggers their processing in the relevant ABAP back end.

You have to enter the key (GUID) of a change list (CL) of the PI directory that contains all PI objects that are required for the WS configuration. You specify this key on the selection screen of the report in the relevant input field. For a mediated communication (that is, a communication using the PI Integration Server (IS)), you require the following PI objects:

- **Sender side:**
  - Sender agreement
  - Sender channel
- **Receiver side:**
  - Receiver agreement
  - Receiver channel

A PI CL is activated in the relevant view after all PI objects are collected in the PI directory. Due to the activation, all the objects are asynchronously transported to the back ends involved, and the generation of the WS configuration is triggered.

Therefore, wait until this activation process is completed. This may take some time. To check the current status of the activation, you can use the cache notification view of the PI directory (choose "Refresh").

If an error occurs at runtime, that is, while the PI messages are exchanged (and results in one of the situations specified above in the section concerning the reasons), it may be useful to repeat the WS configuration in the relevant ABAP back ends.

Therefore, log on to all ABAP back-end systems that are involved in the WS communication (sender, recipient and in the case of the mediated communication, also in the IS client) and start the report SSRV_CPA_TEST in these systems. Specify the CL key or the object ID in each case. If processing is successful, the report does not issue any messages. If an error occurs, the system issues detailed error messages.

You can use the change list key for the mentioned report repeatedly. During each run, the WS configurations that depend on the PI objects in the change list are deleted and regenerated. If you execute the report in an ABAP system that is not involved in the WS communication, the system does not perform any actions.

An additional important reason why you could use the report SSRV_CPA_TEST is that this report enables you to redo the PI cache refresh process part that is performed in an ABAP back end. As a result, you can directly reproduce the actions performed in the current ABAP back end, for example, using the debugger. For a detailed error analysis of the PI cache refresh process in connection with the WS adapter, you require this report.

### 7.1.1.6 Exception “Authentication failed”

**Problem**

An exception is raised when you try to send the message from Consumer to PI system saying that “CX_WS_SECURITY_FAULT: An exception occurred: Authentication failed”.

April 2009 42
Validity

SAP Netweaver PI 7.1 SP06, SAP ECC 6.0 SP 17

Possible Reasons

This may occur because of time skew between the systems involved is greater than 30 seconds or the password of DELAY_LOGON user on Integration Server is changed or this user is locked.

Solution

Reduce the time difference between all the systems to be less than or equal to 30 seconds. Ensure DELAY_LOGON user is not locked.

7.1.1.7 CX_WS_SECURITY_FAULT: An exception occurred: No SAML mapping data in USREXTID

Problem

An exception is raised by the system during SAML authentication which says “CX_WS_SECURITY_FAULT: An exception occurred: No SAML mapping data in USREXTID for...”

Validity

SAP Netweaver PI 7.1 SP06, SAP ECC 6.0 SP 17

Possible Reasons

System cannot find a user (the user who triggers the process i.e. Business user) that matches to Ticket Trust and this user is not maintained in the table VUSREXTID.

Solution

Make sure that you maintain the user mapping by following the link below:

Setting the WS Provider AS ABAP to Accept SAML Token Profiles
Go to transaction SM30 → table view VUSREXTID, type=SA → check if a mapping with the user exists.

7.1.1.8  CX_WS_SECURITY_FAULT: An exception occurred: No trusted certificate for SAML authentication found

**Problem**
You encounter an error while running the SAML scenario saying that "CX_WS_SECURITY_FAULT: An exception occurred: No trusted certificate for SAML authentication found".

**Validity**
SAP Netweaver PI 7.1 SP06, SAP ECC 6.0 SP 17

**Possible Reasons**
Ticket Trust is not maintained. Certificate used for signing the SAML Assertion is not trusted.

**Solution**
Check if the right certificates are used in agreements/logical ports and create trust relationship on AS ABAP for all systems involved.

For more information see SAP Help portal at Setting the WS Provider AS ABAP to Accept SAML Token Profiles

### 7.2 Additional Information

7.2.1 Help Documentation
1. To configure Consumer and Provider ABAP Proxy Service with PI system, see the documentation at [https://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/c00ca32e-f991-2b10-f5be-97114bd2b08f](https://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/c00ca32e-f991-2b10-f5be-97114bd2b08f)
2. For more information on how to configure web service scenarios, see the documentation at [https://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/109aa318-5aa6-2a10-799d-a495383c1e01](https://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/109aa318-5aa6-2a10-799d-a495383c1e01)

7.2.2 Upcoming Features
1. New feature is coming up with SAP PI 7.11 (EHP 1) to separate WSDL access credentials and WS processing security in WS adapter of version in SAP BASIS 7.11.
2. SAML 1.1 assertion issued by a central Security Token Service or Holder of the key authentication method is supported from the release SAP PI 7.11 (i.e. SAP BASIS 7.11)
3. A configurable parameter, for time tolerance between the systems so as to make SAML assertion valid for more than 30 seconds, will be available in EHP 1 for PI 7.1.
7.2.3 **Tips**

1. Go to transaction SICF, browse to default_host → sap → bc → srt → wsd1. Double click on the node, if it says that it is not active, right-click on it and choose ‘activate service’. This helps in activating the wsd1 service.

2. Apply SAP Note 495911 to find out the error code for authentication problems.

3. For ICM errors, check ICM trace (Transaction SMICM, menu Goto → Trace File).

4. For authorization problems, call transaction SU53 to see the failing authorization check.
   
   See Authorization to know which standard roles are required for web service security.