SAP NetWeaver 2004s SPS 4

Security Guide

Web Services Security

Document Version 1.00 – October 24, 2005

SAP AG
Neurottstraße 16
69190 Walldorf
Germany
T +49/18 05/34 34 24
F +49/18 05/34 34 20
www.sap.com

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressively prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.

Documentation in the SAP Service Marketplace

You can find this documentation at the following Internet address:
`service.sap.com/securityguide`

Typographic Conventions

Type Style	Description
Example Text	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
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
EXAMPLE TEXT	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
`Example text`	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
`Example text`	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
`EXAMPLE TEXT`	Keys on the keyboard, for example, *F2* or *ENTER*.

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Additional icons are used in SAP Library documentation to help you identify different types of information at a glance. For more information, see *Help on Help → General Information Classes and Information Classes for Business Information Warehouse* on the first page of any version of *SAP Library*.

Contents

Web Services Security

Purpose

Security is one of the main prerequisites when using Web services in an enterprise. Security measures generally concern both the protection of individual servers through authentication, authorization, and encryption as well as the sealing off of an internal infrastructure using firewalls. Security measures for integrated e-business scenarios must be more diverse since they concern the protection of individual services and data.

Security at transport level can be ensured by means of mechanisms used on the Internet. HTTPS sets up an encrypted connection between the client and the server and is suitable for simple situations – for example, when a client communicates directly with a single server. Every single message is sent via an encrypted channel.

This feature of HTTPS, that each message is encrypted, has two disadvantages.

Firstly, many messages have to be encrypted and decrypted on a single server simultaneously. This can have a negative effect on system performance. Furthermore, the information provided using a Web service is not always confidential and must therefore not always be encrypted.

Secondly, a SOAP interaction is not always a point-to-point connection. More than two SOAP nodes can be involved. The additional intermediate nodes obtain information about actions to be executed from the SOAP header. This is not possible in the case of a complete encryption using HTTPS.

At message level, an encryption and signature concept with fine granularity is possible. Here, not the transport canal but the message itself is protected.

WS Security (OASIS WS Security) is a security model based on SOAP message transfer. WS Security essentially integrates XML Encryption and XML Signature.

There are several security mechanisms available on the SAP J2EE Engine:

- Secure communication using SSL
- Document Security (XML signature and XML encryption)
- Authenticating the client
- Assigning authorizations

Secure Transmission [Page 6]

WS Security [Page 8]

Authentication [Page 11]

Authorization [Page 16]

1 Secure Transmission

To use a Web service, a user (or other client) sends a document to a server using the Simple Object Access Protocol (SOAP), which is then sent over the network using the HTTP protocol. The transmission of the document can either be secured by using HTTP over SSL, or by signing and/or encrypting the SOAP document using OASIS WS Security [Page 8].

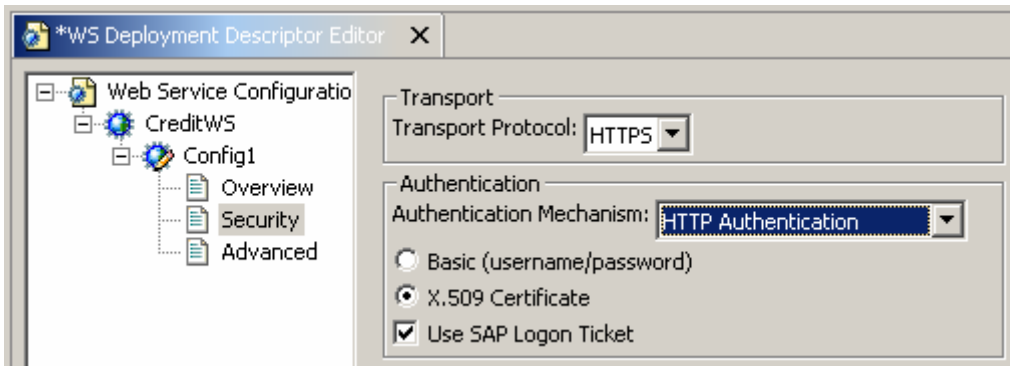
SSL

For transport security, the SSL Protocol is supported by the SAP Web AS and the Web Service Proxy. In this way, all the data for a Web service call can be transmitted between client and server in an encrypted form.

Design-Time Configuration

- Web Service

  To secure transmission using SSL, select HTTPS as the transport protocol in the WS Deployment Descriptor Editor:



  Alternatively, you can proceed to the Web Service Definition, select the feature *Transport Guarantee*, and choose the value *Integrity + Confidentiality*.

- Web Service Proxy

  The Web service called by the proxy must support SSL and have a URL starting with https. Besides entering an URL with **`https://`** no further configuration is needed at design time.

Runtime Configuration

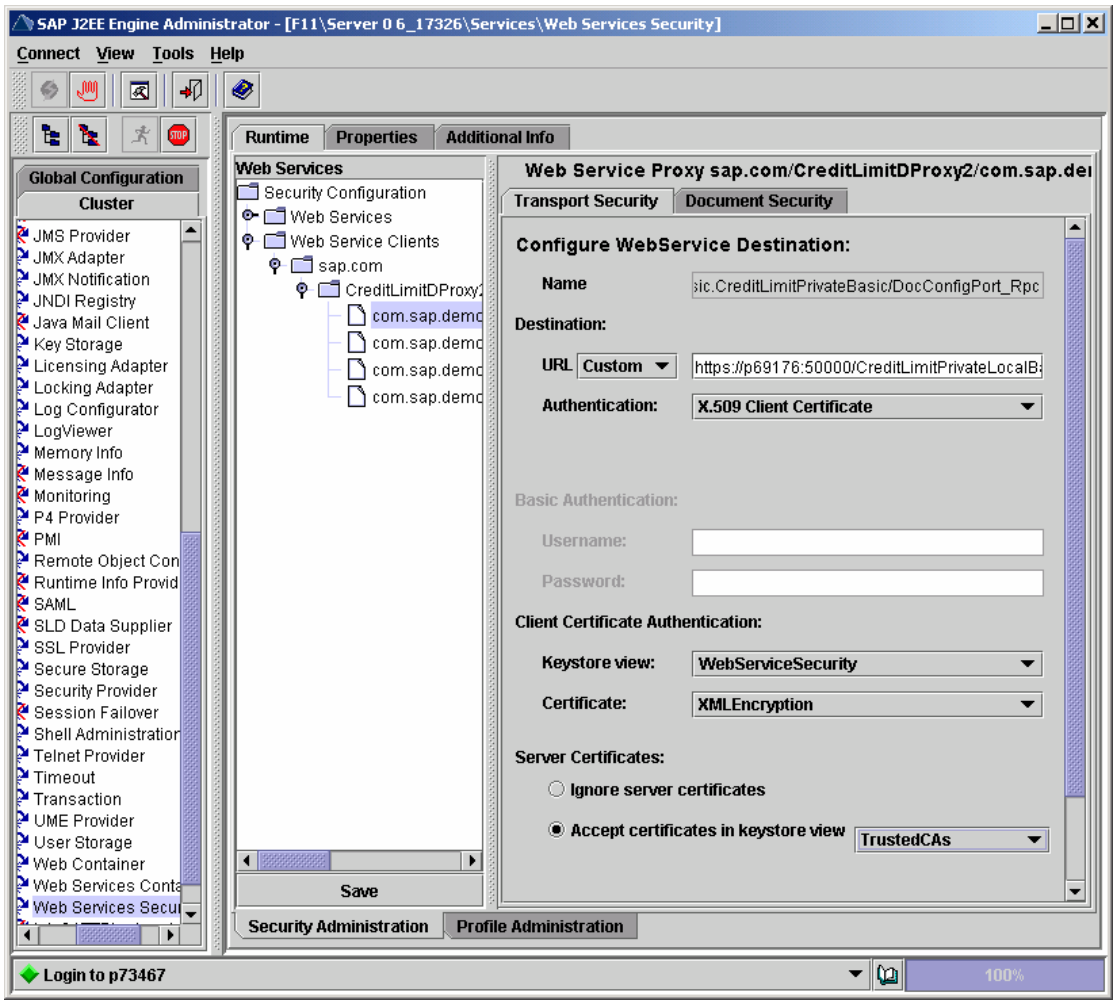
- Web Service

  You have to map client certificates to users (see: Using Client Certificates for User Authentication [SAP Library] and Managing User Certificates [SAP Library] in the Visual Administrator). Make sure that the J2EE Engine has been appropriately configured (see: Configuring the Use of SSL on the SAP J2EE Engine [SAP Library]).

- Web Service Proxy

  As part of establishing an SSL connection, the SSL server certificate is returned. By default, all SSL server certificates are trusted. To limit the accepted SSL server certificates to those issued by certain certificate authorities, the certificates of the certificate authorities must be stored in a keystore view (see: Key Storage Service [SAP Library]).

  In the Visual Administrator, choose the service *Web Service Security*. Choose the client proxy and select the radio button *Accept certificates in keystore view* on *Transport Security* tab.



See also:

Configuring Transport Authentication [Page 12]

Configuring Document Authentication [Page 13]

2 WS Security

WS Security is a standard for securing the SOAP message and does not rely on the Secure Socket Layer Protocol. By using WS Security, SOAP messages passed between the Web service provider and the Web service client are protected by XML digital signatures, XML encryption, timestamps, and security tokens.

At the time of writing, the standardization of WS Security was still in progress. For current information see SAP Note 688983.

WS Security can only be applied to SOAP messages. It is not supported for the *HTTP Get* profile, *HTTP Post* or *SOAP with attachments*. WS Security is only supported by deployable proxies.

XML Signatures

Digital signatures are added to a SOAP document to ensure the integrity and the authenticity of the message. If parts of the message are changed during transport, the signature becomes invalid and the message is rejected by the receiving party. Signatures may be added to client request and the server response. Signatures are always used in combination with a timestamp to prevent replays of the message (both the SOAP:Envelope/SOAP:Body element and the SOAP:Envelope/SOAP:Header/wsse:Security/wsu:Timestamp are signed).

XML Encryption

Encryption is used to protect elements that are sent as part of the SOAP message. For decryption the Key *XMLEncryption* in the keystore view *WebServiceSecurity* is used.

There is limited support for XML encryption in Release 6.40. Decryption of encrypted SOAP documents and encryption of the *Username* security token is supported.

Security Tokens

Besides XML signatures, other credentials used to authenticate the Web service client may be included in the message. The SAP Web AS implementation of WS Security supports the Username security token and the X.509 security token.

To proof the possession of the X.509 certificates used in the X.509 security token, an XML signature using the corresponding private key is required.

Using WS Security

Configuring a Web service to use WS Security settings requires three steps:

1. For each operation in the Web service, select the WS Security template for request and response from the list in the SAP Netweaver Developer Studio. A WS Security Template describes the security (i.e. XML Signature) used to protect the message.
2. For each of the used WS Security templates specified at design time, a profile with runtime configuration settings, such as X.509 certificate data, is required.
3. After creating the WS Security profiles, the profiles need to be assigned to the operations. One profile may be assigned to multiple operations - that is, when the same certificate is to be used for an XML Signature, or different profiles of the same template are used for operations with different XML Signatures.

WS Security Profiles

The following WS Security templates for inbound/outbound messages are available.

Outbound messages (client request, server response):

Security Template:	**Effect:**	**Configuration Parameters:**
Signature	Adds a wsu:Timestamp to the message and signs the elements SOAP:Envelope/SOAP:Header/wsse:Security/wsu:Timestamp and SOAP:Envelope/SOAP:Body.	keystore view, keystore alias for signing key
Username	Adds a SOAP:Envelope/SOAP:Header/wsse:Security/wsse:Username element to the message containing a timestamp, a username and a password.  The password is stored encrypted provided the SAP Java Cryptographic Toolkit [SAP Library] is installed.	username, password
Username + Encryption	Adds a SOAP:Envelope/SOAP:Header/wsse:Security/wsse:Username element to the message containing a timestamp, a username and a password and encrypts the wsse:Username element.	username, password, keystore view and alias of the X.509 certificate (used for XML Encryption)
None	Does not add any security to the message	None

3.1 Configuring Transport Authentication

Inbound messages (client response, server request):

Security Template:	Effect:	Configuration Parameters:
Signature	Verifies the signature over SOAP:Envelope/SOAP:Body and SOAP:Envelope/SOAP:Header/wsse:Security/wsu:Timestamp and checks the validity of the timestamp.	keystore view with the certificates of the trusted certificate authorities. For authentication, the user mapping between X.509 certificate and user provided in the service *Security Provider* is used.
Username	Authenticates the sender using the SOAP:Envelope/SOAP:Header/wsse:Security/wsse:Username element to the message containing a timestamp, a username and a password.	None
Username + Encryption	Decrypts the SOAP:Envelope/SOAP:Header/wsse:Security/wsse:Username element to the message containing a timestamp, a username and a password and encrypts the wsse:Username element.	For decryption the Key *XMLEncryption* in the keystore view *WebServiceSecurity* is used.
None	Does not add any security to the message	None

See also:

Configuring Document Authentication [Page 13]

3 Authentication

Web service clients can authenticate themselves either by using the authentication mechanisms provided by the HTTP protocol such as *HTTP Basic authentication*, or by adding a security token to the WS Security [Page 8] header. Depending on the authentication mechanism, different authentication options are available.

Authentication mechanisms:	**Effect:**
None	Web service client is not authenticated.
Transport Authentication	The Web service client is authenticated using data supplied in the HTTP header or by the SSL protocol. • **Basic Authentication (Username/Password)** Authenticates the caller based on a username and password in the HTTP header. This option is supported for HTTP and HTTPS. • **Strong Authentication (X.509 Client Certificate)** Authenticates the caller using SSL mutual authentication. The caller must provide an SSL client certificate (see: Using Client Certificates for User Authentication [SAP Library]). For further information refer to Configuring Transport Authentication [Page 12].
Document Authentication	The Web service client is authenticated using the security token included in the WS Security header. • **Basic Authentication (Username/Password)** Authenticates the caller based on a username and password in the WS Security SOAP header. • **Strong Authentication (X.509 Client Certificate)** Authenticates the caller based on a digital signature over the SOAP:Body and a timestamp element.  Document authentication supports the transport protocols HTTP and HTTPS. The authentication of standalone proxies is not supported. For further information refer to Configuring Document Authentication [Page 13].

3.1 Configuring Transport Authentication

Basic (Username/Password)

Configuration:	Procedure:
Configuration in the IDE (Web service)	1. Select a configuration of the Web service and open the security configuration. 2. Set the *Authentication Mechanism* to *HTTP Authentication.* 3. Choose the value *Basic (username/password)* to use basic authentication. 4. Select the checkbox *Use SAP Logon Ticket*, if the Web service should also accept SAP Logon Tickets for authentication.
Configuration in the IDE (proxy)	1. Generate a deployable proxy based on the WSDL, after the Web service has been deployed. 2. Open the logical port. 3. Choose the value *Basic (username/password)* to use basic authentication.
Runtime Configuration in the Visual Administrator	Username and password are maintained in the Visual Administrator. 1. Open the Visual Administrator. 2. Select the service *Web Service Security.* 3. In the list of the Web service proxies, select the proxy in the *Web Service Clients* tree. 4. Choose the tab *Transport Security* and set the authentication to *Basic*. Enter username and password.

Strong (X.509 Client Certificate)

Configuration:	Procedure:
Configuration in the IDE (Web service)	1. Select a configuration of the Web service and open the security configuration. 2. Set the transport protocol to *HTTPS.* 3. Set the Authentication Mechanism to *HTTP Authentication.* 4. Choose the value X.509 *Certificate* to use SSL mutual authentication. 5. Select the checkbox *Use SAP Logon Ticket*, if the Web service should also accept SAP Logon Tickets for authentication.
Configuration in the IDE (proxy)	1. After the Web service has been deployed, generate a deployable proxy based on the WSDL. 2. Open the logical port. 3. Choose the value *X.509 Certificate* to use client certificates for authentication.

Configuration:	Procedure:
Runtime Configuration in the Visual Administrator	To use a client certificate for authentication, proceed as follows: 1. Enable SSL and configure the SSL service to use certificates for authentication (see:Configuring the Use of SSL on the SAP J2EE Engine [SAP Library]). 2. Select the service *Web Service Security* in the Visual Administrator. 3. In the list of Web service proxies, select the proxy in the *Web Service Clients* tree. 4. Choose the tab *Transport Security* and set the authentication to *X.509 Client Certificate.* Select a keystore entry to use for authentication.

For standalone proxies, the settings must be made in the security protocol (see Using the Security Protocol [SAP Library]).

See also:

Administration Manual

SAP J2EE Engine Security [SAP Library]

Using Logon Tickets for Single Sign-On [SAP Library]

Development Manual

Secure Transmission [Page 6]

Using the Security Protocol [SAP Library]

3.2 Configuring Document Authentication

Basic (Username/Password)

Configuration:	Procedure:
Configuration in the IDE (Web Service)	1. Select a configuration of the Web service and open the security configuration. 2. Set the *Authentication Mechanism* to *Document Authentication.* 3. Choose the value *Basic (username/password)* to use an wsse:Username token for authentication. 4. In the tab *Document Security* set *Username* for the request and *None* for the response. This will accept a wsse:username security token for authentication. The settings need to be made for each operation.
Configuration in the IDE (proxy)	1. After the Web service has been deployed, generate a deployable proxy based on the WSDL. 2. Open the logical port. 3. Choose the value *Basic (username/password)* to use an wsse:Username token for authentication.

Configuration:	Procedure:
Runtime Configuration in the Visual Administrator	Username and password are maintained in the Visual Administrator. 1. Open the Visual Administrator. 2. Select the service *Web Service Security.* 3. Create an inbound profile: a. Select the tab *Profile Administration.* b. In the tab *Inbound Messages* select *New* to create a new profile. c. Enter a profile name. d. Choose the template *Username.* e. Save the profile. 4. Create an outbound profile: a. Select the tab *Profile Administration* b. In the tab *Outbound Messages* select *New* to create a new profile: c. Enter a profile name. d. Choose the template *Username.* e. Enter username and password. f. Save the profile. 5. Select the proxy in the *Web Service Clients* tree in the list of Web service proxies. 6. In the tab *Document Security* assign the corresponding profile to the operations. 7. In the list of Web services, select the service in the *Web Services* tree. 8. In the tab *Document Security* assign the corresponding profile to the operations.

Strong (X.509 Client Certificate)

Configuration:	Procedure:
Configuration in the IDE (Web service)	1. Select a configuration of the Web service and open the security configuration. 2. Set the *Authentication Mechanism* to *Document Authentication.* 3. Choose the value *X.509 certificate* to use an XML Signature for authentication. 4. In the tab *Document Security* set *Signature* for the request and *None* for the response. This will accept a XML Signature for authentication. The settings need to be made for each operation.

Configuration:	Procedure:
Configuration in the IDE (proxy)	1. After the Web service has been deployed, generate a deployable proxy based on the WSDL. 2. Open the logical port. 3. Choose the value *X.509 Certificate* to use an XML Signature token for authentication.
Runtime Configuration in the Visual Administrator	Inbound and outbound profiles are maintained in the Visual Administrator. 1. Open the Visual Administrator 2. Select the service *Web Service Security* 3. Create an inbound profile: a. Select the tab *Profile Administration.* b. In the tab *Inbound Messages* select *New* to create a new profile. c. Enter a profile name. d. Choose the template *Signature* e. Select a keystore view with trusted root certificates. f. Save the profile. 4. Create an outbound profile: a. Select the tab *Profile Administration.* b. In the tab *Outbound Messages* select *New* to create a new profile. c. Enter a profile name. d. Choose the template *Signature.* e. Select a key from the keystore for signing the message. f. Save the profile. 5. In the list of Web service proxies, select the proxy in the *Web Service Clients* tree. 6. In the tab *Document Security* assign the corresponding profile to the operations. 7. In the list of Web services, select the service in the *Web Services* tree. 8. In the tab *Document Security* assign the corresponding profile to the operations.

See also:

WS Security [Page 8]

4 Authorization

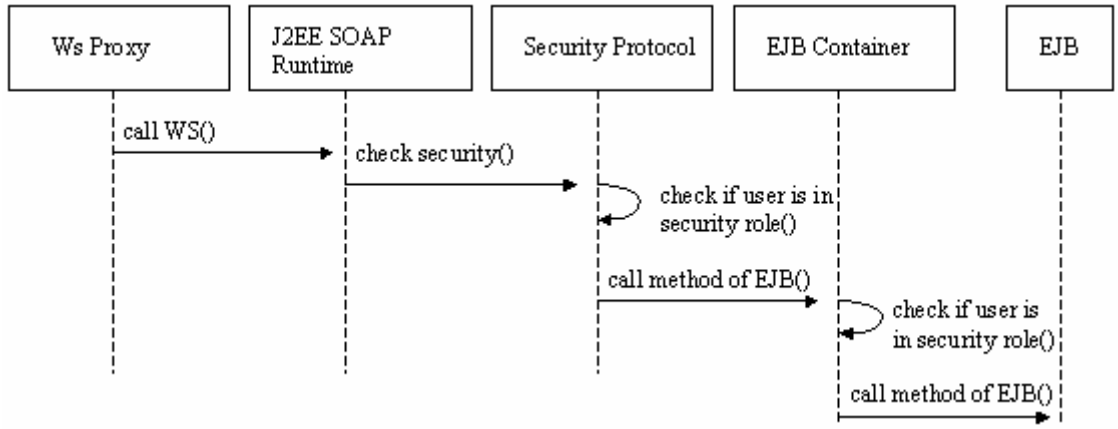
The authorization concept used for Web services depends on the type of Web service:

EJBs

Web service operations of an EJB can be protected by roles. The roles can be checked in one of two places: Either for the virtual interface or – in accordance with the J2EE specification – for the methods of the EJB. It is also possible to execute the check at both places.

- **Authorization check for the virtual interface:** Calling methods of virtual interfaces can be limited to users with one or several roles. If there are several virtual interfaces for an EJB (possibly with different predefined parameters), different roles can be checked for each virtual interface. This authorization check takes place for all WS calls.
- **Authorization check for the EJB methods**: The roles are checked using the EJB container - that is, the check is executed for direct calls to EJB (P4 protocol) as well as for WS calls.

The security roles are checked in the server in the SOAP runtime. The authorization check for the methods of the virtual interface takes place in the security protocol in the SOAP runtime. The EJB methods are checked in the EJB container.



To limit access to the operations of an EJB, proceed as follows:

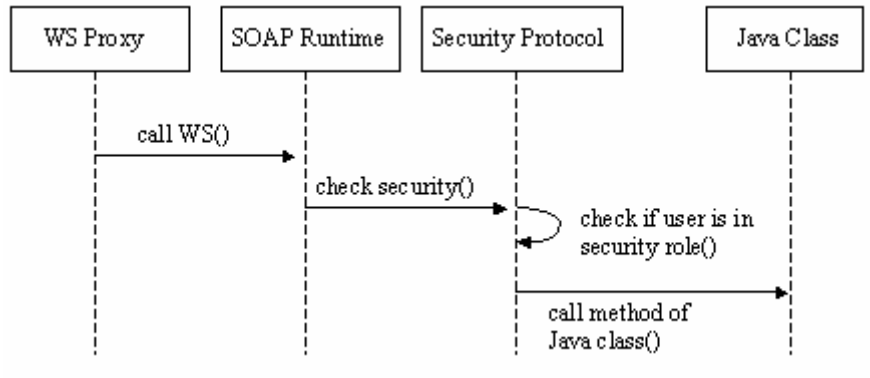
- Choose the *Features* tab in the Web Service Definition. Choose *Authorization* and *Select Feature.*
- Open the *ejb-jar.xml* descriptor. Choose the *Assembly tab* and add security roles.
- Configure the authorization check for virtual interface authorizations in the WS Deployment Descriptor. Choose a configuration under *Web Service Configurations* in the Web service perspective. In the tree under the configuration name choose *Security*.
- Configure authorization checks for the operations.
- Map the security roles to users in the Visual Administrator (see: Mapping Users and Groups [SAP Library]).

  To maintain the roles in the Visual Administrator choose *Security Provider.* Under *Components* search for *providername/EAR project*JAR-File.* (The name of the provider can be changed in the file application.xml).

Java Classes

The authorization check for Java classes takes place for the virtual interface methods. In this way, access to methods exposed as Web services are limited through the use of J2EE security roles.

The security roles are checked on the server in the SOAP runtime. Before a Java class method is called, the system checks in the security protocol whether the user is assigned to a particular security role.



To limit access to the operations of a Java class, proceed as follows:

- Configure the authorization check for virtual interface authorizations in the WS Deployment Descriptor. Choose a configuration under *Web Service Configurations* in the Web service perspective. In the tree under the configuration name choose *Security Roles* to add security roles.
- Configure authorization checks for the operations under the node *Security* of the Web service configuration.
- Map the security roles to users in the Visual Administrator (see: Mapping Users and Groups [SAP Library]).

  To maintain the roles in the Visual Administrator choose *Security Provider.* Under *Components* search for *providername/Java project*Name of Web service_Name of configuration.*

See also:

Security Roles Management [SAP Library]