SAP Enterprise Portal Security Guide

SAP EP 6.0 SP1

Version 1.3

Copyright

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Typographic Conventions

Type Style	Description
Example text	Words or characters that appear on the screen. These include field names, screen titles, pushbuttons as well as menu names, paths and options. Cross-references to other documentation.
Example text	Emphasized words or phrases in body text, titles of graphics and tables.
EXAMPLE TEXT	Names of elements in the system. These include report names, program names, transaction codes, table names, and individual key words of a programming language, when surrounded by body text, for example, SELECT and INCLUDE.
`Example text`	Screen output. This includes file and directory names and their paths, messages, source code, names of variables and parameters as well as names of installation, upgrade and database tools.
`EXAMPLE TEXT`	Keys on the keyboard, for example, function keys (such as F2) or the ENTER key.
`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. Pointed brackets indicate that you replace these words and characters with appropriate entries.

SAP Enterprise Portal Security Guide

SAP Enterprise Portal offers users a single point of access to all applications, information, and services needed to accomplish their daily tasks. Links to back-end and legacy applications, self-service applications, company intranet services, and Internet services are all readily available in the user's portal. Because the borders between company intranets and the Internet are blurring, comprehensive security is vital to protect the company's business.

In this guide you will find the following security-related topics:

- The section on authentication [Page 7] describes how authentication is handled in the portal and how to configure the portal for anonymous access.
- Authorizations [Page 30] provides an overview of the authorization concepts in the portal and points you to where you can find more details.
- The following section outlines the different variants of Single Sign-On [Page 31] available in the portal and describes how to set up these variants.
- Finally the section on secure communications [Page 47] provides an overview of the communication channels used in a typical SAP Enterprise Portal installation. It also covers Secure Sockets Layer (SSL) communication with an LDAP directory.

For information on user management, see the following sections in *SAP Enterprise Portal Administration Guide*:

- *SAP Enterprise Portal Administration Guide → Portal → User Administration*
- *SAP Enterprise Portal Administration Guide → Portal → System Administration → User Management Configuration*

Authentication

Authentication provides a way of verifying the user's identity before he or she is granted access to the portal. Once the user has been authenticated, he or she is issued a SAP logon ticket that allows him or her to access all the applications, information and services in SAP Enterprise Portal using Single Sign-On. Since many of these applications may contain sensitive data, it is imperative that the user in question can be identified and this identity authenticated.

The process of authentication is based on each user having a unique set of credentials for gaining access. For example, with user ID and password authentication, the authentication server compares a user's authentication credentials with other user credentials stored in a data repository. If the credentials match, the user is granted access to the Enterprise Portal. Otherwise, the authentication fails and portal access is denied.

In the portal, authentication is defined using authentication schemes [Page 8] which are assigned to iViews. Users log on to the portal with a specific authentication scheme and this is stored in the user's logon ticket. If a user needs to access an iView which requires a stronger authentication scheme, he or she must re-authenticate as specified by the stronger authentication scheme.

The portal offers the following authentication mechanisms:

- Authentication with user ID and Password
  - Form-based logon (default authentication method)
  - Basic Authentication
- Authentication with X.509 client certificates [Page 19]
- Authentication using external mechanisms

- Windows authentication [Page 21]
- Authentication Using Web Access Management Products [Page 23]

In addition, it is possible to configure the portal for anonymous access [Page 24].

To log on to the portal, users must enter the full URL in the browser including the fully qualified domain name, otherwise the browser will not get the correct SAP logon ticket. If the portal is running in the intranet only, you can configure your Web server to change a host name to a full URL.

If you want the logon screens to reflect your corporate design and include your company logo, you can customize them as required. For more information, see Customizing the Logon Screens [Page 29].

Authentication Schemes

Definition

An authentication scheme is a definition of what is required for an authentication process. This includes:

- Type of information used to compute user's identity. For example, user ID and password, client certificate, and so on.
- How user data is checked. For example, against an LDAP directory or a SAP R/3 System.
- Validity of user logon, that is, the amount of time after which a user has to log on again.
- Priority, allowing authentication schemas to be ordered.

Use

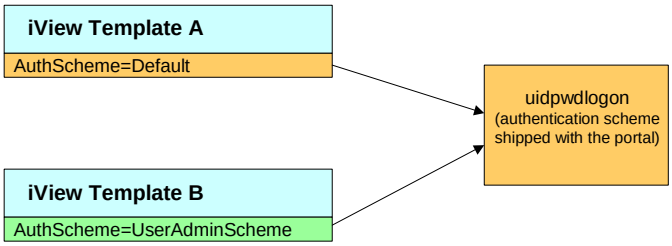
Authentication schemes allow you to enforce different authentication mechanisms for different content. Each iView is assigned an authentication scheme and only users that have logged on successfully with that authentication scheme or one with a higher priority can access the iView.

In addition, authentication schemes enable pluggable authentication. You can easily ‘plug in’ additional authentication schemes into the portal using modules that adhere to the *Java Authentication and Authorization Service* (JAAS) standard.

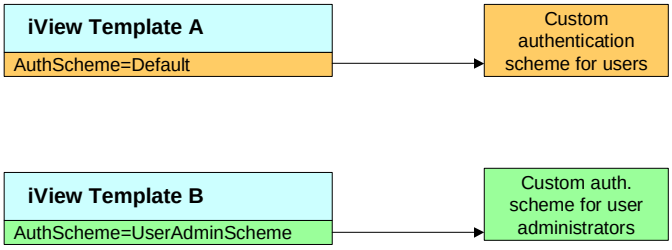
Integration

SAP Enterprise Portal is shipped with a set of authentication schemes [Page 16]. Each shipped iView template is assigned a reference to an authentication scheme. Initially all references to authentication schemes point to the same authentication scheme (Default). If you have special authentication requirements, you can define custom authentication schemes and then change the configuration of the portal so that the references point to your custom authentication schemes. This allows you to change the authentication schemes without having to modify the iViews or iView templates. The following diagram illustrates this concept:

Initial Configuration of Portal



(Optional) Custom Configuration of Portal



For details on changing the references to authentication schemes, see Defining References to Authentication Schemes [Page 13].

For details on defining new authentication schemes, see Defining an Authentication Scheme [Page 10].

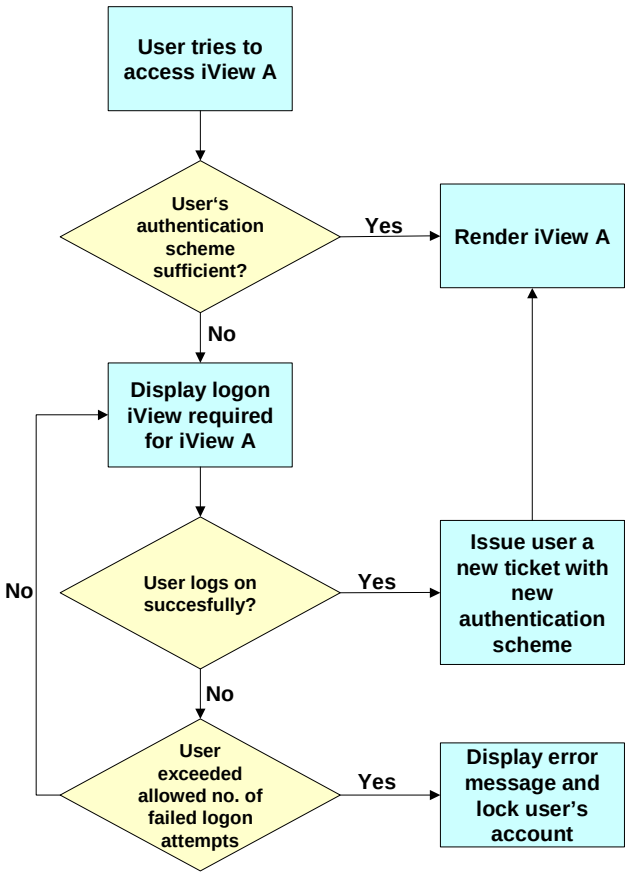
What Happens When a User Logs on to the Portal

When users launch SAP Enterprise Portal, they are required to log on with the authentication scheme that corresponds to the iViews on the first displayed page. If the users satisfy the authentication requirements for the authentication scheme, this information is stored in their logon ticket. If users try to access an iView that requires a ‘stronger’ authentication scheme, the users will have to re-authenticate themselves and will be issued a new logon ticket with the new authentication scheme in it.

For example: The authentication scheme in which users are logged on with a client certificate has a higher priority than an authentication scheme in which users are logged on with user ID and password. This means that users logged on with a client certificate can access all iViews that require an authentication scheme in which users are logged on with user ID and password.

If a user that is logged on with user ID and password tries to access an iView that requires authentication with a client certificate the user will have to re-authenticate and provide a client certificate.

The following diagram illustrates the flow of authentication.



SAP Enterprise Portal is shipped with a set of default authentication schemes. In addition, you can define you own authentication schemes to suit your company's individual requirements.

Defining an Authentication Scheme

Use

You can define custom authentication schemes if your specific requirements are not covered by the authentication schemes shipped with the portal. You define authentication schemes in the file authschemes.xml [Page 102].

Procedure

Open the file authschemes.xml [Page 102] for modifying as described in Changing the authschemes.xml File [Page 15].

Here is an example of the contents of this file:

```
<document>
    <authschemes>
        <!--  authschemes, the name of the node is used -->
        <authscheme name="uidpwdlogon">
            <!-- multiple login modules can be defined -->
            <loginmodule>
                <loginModuleName>
                    com.sap.security.core.logon.imp.CertLoginModule
                </loginModuleName>
                <!-- specifying whether this LoginModule
                    is REQUIRED, REQUISITE, SUFFICIENT,
                    or OPTIONAL -->
                <controlFlag>SUFFICIENT</controlFlag>
                <options></options>
            </loginmodule>
            <loginmodule>

                <loginModuleName>
                    com.sap.security.core.logon.imp.DefaultLoginModul
e
                </loginModuleName>
                <controlFlag>REQUISITE</controlFlag>
                <options></options>
            </loginmodule>
            <priority>20</priority>
            <!-- the frontendtype TARGET_FORWARD = 0,
                TARGET_REDIRECT = 1, TARGET_JAVAIVIEW = 2 -->
            <frontendtype>2</frontendtype>
            <!-- target object -->
            <frontendtarget>com.sap.portal.runtime.logon.certlogon</f
rontendtarget>
        </authscheme>


        <authscheme name="certlogon">
            <!-- multiple login modules can be defined -->
            <loginmodule>
                <loginModuleName>
                    com.sap.security.core.logon.imp.CertLoginModule
                </loginModuleName>
                <!-- specifying whether this LoginModule
                    is REQUIRED, REQUISITE, SUFFICIENT,
                    or OPTIONAL -->
                <controlFlag>REQUISITE</controlFlag>
                <options></options>
            </loginmodule>
            <priority>21</priority>
            <!-- the frontendtype TARGET_FORWARD = 0,
                TARGET_REDIRECT = 1, TARGET_JAVAIVIEW = 2 -->
            <frontendtype>2</frontendtype>
            <!-- target object -->
            <frontendtarget>com.sap.portal.runtime.logon.certlogon</f
rontendtarget>
        </authscheme>
    </authschemes>

    <!--  References for Authentication Schemes,
        this section must be after authschemes -->
    <authscheme-refs>
        <authscheme-ref name="default">
            <authscheme>uidpwdlogon</authscheme>
        </authscheme-ref>
```

```
    </authscheme-refs>
</document>
```

To define an authentication scheme (`authscheme`), you need to provide the following information:

- Login module
- Priority
- Frontend type
- Frontend target

These are described in more detail below.

Login Module

You can define one or more login modules for an authentication scheme. You can implement custom login modules that implement the *LoginModule* interface of the *Java Authentication and Authorization Services* (JAAS). For details on *LoginModule*, see Sun's Java documentation (http://java.sun.com).

In the `loginModuleName` tag, enter the fully qualified class name of a class that implements the *LoginModule* interface.

```
<loginModuleName>
    com.sap.security.core.logon.imp.DefaultLoginModule
</loginModuleName>
```

In the `controlFlag` tag, enter a flag value which controls what happens as authentication proceeds through the list of login modules. These flags have the same meanings as those described by Sun:

Value of controlFlag	**Meaning**
`Required`	The LoginModule is required to succeed. If it succeeds or fails, authentication still continues to proceed down the list of login modules.
`Requisite`	The LoginModule is required to succeed. If it succeeds, authentication continues down the list of login modules. If it fails, control immediately returns to the application (authentication does not proceed down the list of login modules).
`Sufficient`	The LoginModule is not required to succeed. If it does succeed, control immediately returns to the application (authentication does not proceed down the list of login modules). If it fails, authentication continues down the list of login modules.
`Optional`	The LoginModule is not required to succeed. If it succeeds or fails, authentication still continues to proceed down the list of login modules.

The overall authentication succeeds only if all Required and Requisite LoginModules succeed. If a Sufficient LoginModule is configured and succeeds, then only the Required and

Requisite LoginModules prior to that Sufficient LoginModule need to have succeeded for the overall authentication to succeed. If no Required or Requisite LoginModules are configured for an application, then at least one Sufficient or Optional LoginModule must succeed.

In the options tag, you can optionally provide options which are passed to the login module. The options are defined by the LoginModule itself and control the behavior within it. To define options, you use a simple `name=value` syntax. For example:

```
<options>system=“ABC“</options>
```

Priority

The priority of an authentication must be a positive integer.

```
<priority>20</priority>
```

The higher the integer, the higher the priority of the authentication scheme. Each iView is assigned an authentication scheme and only users that have logged on successfully with that authentication scheme or one with the same or a higher priority can access the iView.

For example, an authentication scheme that requires the user to authenticate using user ID and password has a priority of 10. An authentication scheme that requires the user to authenticate using a client certificate has a priority of 20. If a user has authenticated himself with a client certificate (priority 20) and then tries to access an iView that requires authentication with user ID and password (priority 10) he will not need to re-authenticate himself.

Frontend type

In the Enterprise Portal the frontend type must always be 2.

Frontend target

The frontend target defines which iView is to be launched when a user’s session does not satisfy the required authentication scheme. Whereas the login module defines how the user is authenticated, the frontend target defines the user interaction that needs to take place to gather the required information.

In addition you may want to define a reference to an authentication scheme. For details, see Defining References to Authentication Schemes [Page 13].

When you are finished editing `authschemes.xml`, save the file and proceed as described in Changing the authschemes.xml File [Page 15].

Result

You have defined a custom authentication scheme and can assign it to iView templates or iViews. For details, see Assigning an Authentication Scheme to an iView [Page 14].

Defining References to Authentication Schemes

Use

A reference to an authentication scheme is a ‘pointer’ to an authentication scheme. All iViews templates shipped with SAP Enterprise Portal have a property that contains a reference to an authentication scheme. By changing what the reference points to (that is, by modifying a reference to an authentication scheme), you can change the authentication scheme for a whole set of iViews and iView templates without having to change the property in each individual iView or iView template.

Procedure

Open the file authschemes.xml [Page 102] for modifying as described in Changing the authschemes.xml File [Page 15].

The first part of this file contains a list of authentication schemes. At the end of the file you can define references to authentication schemes. The following is an example:

```
<!--  References for Authentication Schemes,
    this section must be after authschemes -->
<authscheme-refs>
    <authscheme-ref name="default">
        <authscheme>uidpwdlogon</authscheme>
    </authscheme-ref>
</authscheme-refs>
```

In the above example, the reference `default` points to the authentication scheme called `uidpwdlogon` that is defined in the same file. All iView templates that are assigned to the authentication scheme reference `default` require the `uidpwdlogon` authentication scheme. By changing `uidpwdlogon` to `basicauthentication`, for example, all the iView templates that are assigned to `default` now require the `basicauthentication` authentication scheme.

When you are finished editing `authschemes.xml`, save the file and proceed as described in Changing the authschemes.xml File [Page 15].

Assigning an Authentication Scheme to an iView

Use

All iViews shipped with SAP Enterprise Portal have an authentication scheme assigned to them. You can change this authentication scheme in the properties of the iView.

Procedure

Use one of the following procedures to assign an authentication scheme to an iView.

Assigning an authentication scheme to an iView or iView template

1. In the portal, choose *Content Administration* → *Portal Content*.

   The Portal Content Studio is displayed.

2. In the Portal Catalog on the left, navigate to the iView that you want to change.

3. Right click on the iView and choose *Edit*.

   The property editor of the iView is opened in the editing area on the right.

4. Change the *Authentication Scheme* property to either:
    - the name of an authentication scheme
    - a reference to an authentication scheme

   The default value for this property is *default*.
5. Save your changes.

Assigning an authentication scheme to a portal component before uploading it into the portal

1. In the configuration section of the `portalapp.xml` file of the portal component, set the property `AuthScheme` to either
    - the name of an authentication scheme
    - a reference to an authentication scheme

```
<component-profile>
    <property name="ForcedRequestCountry" value="">
        <property name="personalization" value="none"/>
    </property>
    <property name="ForcedRequestLanguage" value="en">
        <property name="personalization" value="none"/>
    </property>
    <property name="AuthScheme"
value="basicauthentication"/>
</component-profile>
```

2. Save the file.
3. Upload the portal component into the portal.

Changing the authschemes.xml File

Use

Any changes you make to the authschemes.xml [Page 102] file must be uploaded into the Portal Content Directory (PCD) for them to take effect.

`authschemes.xml` is loaded into the PCD from `com.sap.instancedir\ume` on initial startup of the portal (`com.sap.instancedir` is a system property of SAP J2EE Engine). After it is loaded into the PCD, it is renamed to `authschemes.xml.bak`. To force it to be reloaded into the PCD, the `.bak` must be removed. It is then reloaded on restart of the portal. This is described below.

Procedure

1. Browse to the directory `<SAP_J2EE_Engine_installation_directory>\ume`.
2. Rename `authschemes.xml.bak` to `authschemes.xml`.
3. Modify `authschemes.xml` as required.
4. Restart the Java application server.

Result

The modified version of `authschemes.xml` is loaded into the PCD.

Authentication Schemes Shipped with SAP Enterprise Portal

The following authentication schemes are shipped with SAP Enterprise Portal:

Name of Authentication Scheme	Description	Referenced by
uidpwdlogon	Allows authentication with client certificates, but does not require it. If the client does not present a certificate the user logs on with user ID and password. • If the portal is set up for HTTPS with client authentication, this authentication scheme allows authentication using client certificates. • If the portal is set up for HTTPS without client authentication or for HTTP, this authentication scheme requires form-based logon with user ID and password. Requires additional configuration on the Java Server. For details, see uidpwdlogon: Configuring SAP J2EE Engine [Page 17].	default, UserAdminScheme
certlogon	Requires authentication using client certificates.	
basicauthentication	Uses the Basic Authentication feature of the HTTP protocol.	
header	Allows authentication using external Web access management products.	
guest	Allows automatic logon with a named anonymous user. A logon ticket is issued.	
anonymous	Not listed in `authschemes.xml`. Provides a very basic form of anonymous logon. A logon ticket is **not** issued.	

uidpwdlogon: Configuring SAP J2EE Engine

Use

The authentication scheme *default/uidpwdlogon* allows three types of authentication in parallel:

- User authentication with client certificates on a portal set up for Secure Sockets Layer (SSL) with mutual authentication (client and server present a certificate).
- User authentication with user ID and password on a portal set up for SSL with server authentication (only server presents a certificate).
- Authentication with user ID and password on a portal not set up for SSL. This portal is accessed via HTTP only.

To allow all three types of authentication in parallel, you need to configure SAP J2EE as described in this procedure. For full details on how to configure SAP J2EE Engine for SSL, see the SAP J2EE Engine documentation.

Procedure

1. On the portal server, run the file <SAP_J2EE_Engine_installation_directory>\admin\go to start the administrator tool of SAP J2EE Engine.
2. In the list of services on the left, choose *SSL*.
3. Choose the *Trusted Certificates* tab.
4. Select *New Sockets*.
5. Make sure that the list of trusted Certification Authorities (CAs) is empty.

   If this list contains CAs and no client certificate is sent to the server, SAPJ2EE Engine does not establish a HTTPS connection.
6. Select *Require client authentication*.

   If this option is not selected, the browser does not send a certificate.
7. Select *Active Sockets* and repeat steps 5 and 6.

uidpwdlogon: Disabling Certificates Being Mapped at Logon

Use

If you are using the authentication scheme *uidpwdlogon* and the portal is configured for Secure Sockets Layer (SSL) with client authentication (and the administrator has not mapped the users' certificates), users have to map their certificate to their user ID the first time they log on to the portal by entering their user ID and password.

In some cases users may not wish to map their certificate to a user ID, for example, if they need to log on as two different users from the same client. In this case they can click on the link to log on with user ID and password in the logon page. This takes them to a different logon page where they can log on with user ID and password and their certificate is not mapped to a user.

To avoid users having to navigate from one logon page to another, it is also possible to change the authentication scheme *uidpwdlogon* so that certificate mapping is disabled. The following procedure describes how to disable certificate mapping.

Procedure

1. Download the file authschemes.xml [Page 102] from the Portal Content Directory (PCD).
2. In the authentication scheme `uidpwdlogon`, comment out the login module `com.sap.security.core.logon.imp.CertPersisterLoginModule`.

   This login module is responsible for mapping certificates to users.
3. In the authentication scheme `uidpwdlogon`, change the frontend target to `com.sap.portal.runtime.logon.default`.

   The section in the file on the authentication scheme `uidpwdlogon` should look as follows:

```
<authscheme name="uidpwdlogon">

    <loginmodule>
        <loginModuleName>com.sap.security.core.logon.imp.CertLo
ginModule</loginModuleName>
        <controlFlag>SUFFICIENT</controlFlag>
        <options></options>
    </loginmodule>

    <loginmodule>
        <loginModuleName>com.sap.security.core.logon.imp.Defaul
tLoginModule</loginModuleName>
        <controlFlag>REQUISITE</controlFlag>
        <options></options>
    </loginmodule>

<!--    <loginmodule>
        <loginModuleName>com.sap.security.core.logon.imp.CertPe
rsisterLoginModule</loginModuleName>
        <controlFlag>OPTIONAL</controlFlag>
        <options></options>
    </loginmodule>
-->
    <priority>20</priority>
    <frontendtype>2</frontendtype>
    <frontendtarget>com.sap.portal.runtime.logon.default</front
endtarget>
</authscheme>
```

4. When you are finished editing `authschemes.xml`, upload the file into the PCD.

Result

When users log on to the portal and send a client certificate, the client certificate is ignored, and users log on with user ID and password.

Authentication Using Client Certificates

Use

If you require a high level of security, you can use certificate-based authentication through the Secure Sockets Layer (SSL) protocol in your Enterprise Portal. The actual authentication takes place by the SSL protocol between Web browser and Web server, during the so-called SSL handshake. SSL authentication and X.509 certificates use Internet standard technology that provides a higher level of security and eliminates the need for passwords altogether.

Certificate-based authentication provides a high level of security for applications with highly sensitive company data. However, it also requires the company to invest in a public key infrastructure (PKI).

The portal maps client certificates to portal users. The first time users log on with a client certificate, they must enter their user ID and password. The portal uses this information to map the certificate. Alternatively, administrators can map certificates to a user.

The authentication schemes *uidpwdlogon* and *certlogon* allow for authentication with client certificates. For more information, see Authentication Schemes [Page 8] and Authentication Schemes Shipped with SAP Enterprise Portal [Page 16].

Prerequisites

- Users have obtained valid X.509 client certificates as part of a public key infrastructure (PKI) and have imported them into their Web browsers.

  The role of the PKI is to verify the identity of certificate owners and to issue, validate, renew, and revoke certificates. If you use X.509 client certificates for authentication, then you need access to a PKI. You can either establish your own PKI or you can rely on a Trust Center for these tasks.

- The browser and portal Web server are configured to communicate using SSL. See the Web server documentation for detailed instructions.

  If you are using the Web server functions of the SAP J2EE Engine, you can find instructions in the document Configuring the Use of SSL on the SAP J2EE Engine [Page 103].

- The portal Web server is configured to trust the Certification Authority (CA) that issued the user certificates. See the Web server documentation for detailed instructions.

  If you are using the Web server functions of the SAP J2EE Engine, you can find instructions in the document Configuring the Use of SSL on the SAP J2EE Engine [Page 103] → *Configuring the Use of Client Certificates for Authentication*.

- The portal Web server is configured to accept client certificates. See the Web server documentation for detailed instructions.

  If you are using the Web server functions of the SAP J2EE Engine, you can find instructions in the document Configuring the Use of SSL on the SAP J2EE Engine [Page 103] → *Configuring the Use of Client Certificates for Authentication*.

- Users must log on to the portal using https.

Activities

- In user management properties, make sure that the property `ume.logon.allow_cert` is set to `TRUE`.

  `ume.logon.allow_cert=TRUE`

  `For more information on setting user management properties, see` *Enterprise Portal Administrator Guide → Portal → User Management Configuration → User Management Properties.*

- Each user's client certificate must be mapped to his portal user ID. There are two options for this. Either the administrator maps users' certificates to portal user IDs, or each user maps his or her certificate the first time he or she logs on to the portal that has been set up for certificates. For more information on mapping certificates, see *Enterprise Portal Administrator Guide → Portal → User Administration → User Management Administration Console → Mapping Client Certificates to Users.*

Result

Users log on to the portal using https.

When a user accesses an iView that requires certificate logon (the authentication scheme of the iView is `certlogon`), the browser must present a certificate to authenticate the user. If the presented certificate has not been mapped to a user yet, the user will be prompted for user ID and password. After the user enters user ID and password, the certificate is mapped to that user. The Portal Server authenticates the user and issues an SAP logon ticket to the user. The next time the user logs on with a certificate, he or she will no longer need to enter user ID and password.

Windows Authentication

Use

In Windows authentication, authentication of users is delegated to the operating system. You can use the following Windows authentication methods:

Basic Authentication: This authentication mechanism is based on the Basic Authentication feature of the HTTP protocol. The portal user enters his or her existing Windows user name and password into the browser dialog box. The Windows Domain Controller then authenticates the user. This mechanism is typically deployed when the enterprise portal is accessible from the extranet. With this authentication method, the password is transmitted unencrypted, so you should ensure that all connections use SSL.

If you are using basic authentication, we strongly recommended that you set up the browser and portal Web server to communicate using Secure Sockets Layer (SSL). Otherwise users‘ credentials will be transmitted in clear text.

Integrated Windows authentication (previously known as NT Challenge/ Response): If the enterprise portal is implemented as an intranet portal only, a previously successful logon to the Windows operating system can be reused for automatically logging the user on to the portal. This authentication mechanism is based on Windows security. The user is not required to reenter his Windows authentication credentials again. But in order for this to work, the client must use a Microsoft Internet Explorer browser and be within the same Windows domain as the Portal Server.

Configuring Windows Authentication

Prerequisites

- You require a Microsoft Internet Information Server (IIS) as portal Web server. For Windows authentication you cannot use the native Web server of SAP J2EE Engine. It is possible to have the portal running on a UNIX machine and the IIS running on a Windows machine.
- Users must have the same Windows user IDs and portal user IDs.

Procedure

Install the SAP J2EE Engine ISAPI module in the IIS

You can find the ISAPI module on your portal installation at `<J2EE_Engine_installation>\tools\lib\IIS_module`. For documentation on how to install the module, see the *SAP J2EE ISAPI Module Installation Guide* at the same location.

Read also SAP Note 578554 for additional configuration. In particular, make sure that you do the following:

- In `SAPJ2EE.ini`, enable the connection pool size property:

  ```
  connection.pool.size = 100
  ```

- In `SAPJ2EE.ini`, specify the URL mapping:

  ```
  url.mapping       = http: /irj --> http://localhost:8100 \
                    https: /irj --> https://localhost:8443
  ```

Configure Microsoft Internet Information Server (IIS)

1. Start the IIS.
2. Navigate to *Default Web Site/Scripts/SAPJ2EE.dll*.
3. Right click on *SAPJ2EE.dll* and choose *Properties*.

   The properties of *Default Web Site* are displayed.
4. On the *File Security* tab, choose *Edit* under *Anonymous access and authentication control*.

   The *Authentication Methods* dialog box is displayed.
5. Select either *Basic Authentication* or *Integrated Windows authentication* depending on which authentication method you require.
6. Restart the IIS.

Define a New Authentication Scheme

Define a new authentication scheme by adding the following excerpt to authschemes.xml. You can find this file at <SAP_J2EE_Engine_installation_directory>\ume.

```
<authscheme name="windows">
    <loginmodule>
        <loginModuleName>
            com.sap.security.core.logon.imp.WindowsLoginModule
        </loginModuleName>
        <controlFlag>REQUISITE</controlFlag>
    </loginmodule>
    <priority>15</priority>
    <frontendtype>2</frontendtype>
    <frontendtarget>
        com.sap.portal.runtime.logon.certlogon
    </frontendtarget>
</authscheme>
```

Configure the Portal

In the portal you have two options for configuring Windows authentication:

- On a component level
- For the complete portal

On a Component Level

Change the Authentication Scheme property of the component to the authentication scheme windows that you just defined. For more information, see Assigning an Authentication Scheme to an iView [Page 14].

Users can launch the component using the direct URL for that component and are authenticated with Windows authentication.

For example, users can call up the self-registration iView directly using the following URL:

http://<server>:<IIS_port>/irj/servlet/prt/portal/prtroot/usermanagementadmin.SelfReg

For the Complete Portal

Change the reference of the default authentication scheme to the authentication scheme windows that you just defined. For more information, see Defining References to Authentication Schemes [Page 13].

```
<authscheme-refs>
    <authscheme-ref name="default">
```

```
            <authscheme>uidpwdlogon</authscheme>
        </authscheme-ref>
    </authscheme-refs>
```

Result

When a user launches the portal or an iView, his or her user credentials are authenticated against the Windows Domain Controller. The portal issues a logon ticket for the authenticated user.

The URL to launch the portal or iView must contain the port number of the IIS and not of SAP J2EE Engine.

Authentication Using Web Access Management Products

Use

The enterprise portal allows you to delegate user authentication to external Web Access Management (WAM) products such as Netegrity SiteMinder. This is useful if, for example, you are already using an external product to protect other resources in your company, or if you wish to use authentication mechanisms that are not directly supported by the portal, such as token cards or biometrics.

Integration

The portal provides an authentication scheme called `header` containing a JAAS login module that reads a user ID from the HTTP header variable.

Authentication with an external WAM product works as follows: The WAM product authenticates the portal user and returns an authenticated user ID to the portal as part of the HTTP header. The portal that is configured for external authentication compares this returned user ID against the user data sources and grants access to the portal upon finding a match. The portal does not perform any additional authentication of the user. The user must exist in the UME user data sources.

An SAP logon ticket is still generated and stored in the user's browser to enable Single Sign-On in the portal.

Prerequisites

- To use a WAM product with the JAAS login module provided by the portal, you must have an external Web server in front of the portal. All requests must pass through the external Web server.
- The user ID that the WAM product returns in the HTTP header must exist in the user management data sources of the portal.

Activities

The exact steps for setting up authentication using a WAM product depends on the product you use. In all cases you will need to perform the following steps:

1. Open `authschemes.xml` for editing as described in Changing the authschemes.xml File [Page 15].

2. In `authschemes.xml`, change the reference of the default authentication scheme to the authentication scheme `header`. For more information, see Defining References to Authentication Schemes [Page 13].

```
<authscheme-refs>
    <authscheme-ref name="default">
        <authscheme>header</authscheme>
    </authscheme-ref>
</authscheme-refs>
```

3. In `authschemes.xml` in the authentication scheme header, specify the name of the HTTP header variable in which the user ID is supplied by the WAM product.

For example, by default Netegrity SiteMinder supplies the user ID in the HTTP header variable called `HTTP_SM_USER`. In this case, you need to change `authschemes.xml` as follows:

```
<authscheme name="header">
    <loginmodule>
        ...
        <options>Header=HTTP_SM_USER</options>
    </loginmodule>
    <priority>5</priority>
    <frontendtype>2</frontendtype>
    <frontendtarget>com.sap.portal.runtime.logon.header</fr
ontendtarget>
</authscheme>
```

As an alternative to specifying the HTTP header variable in `authschemes.xml`, you can define it using the `ume.logon.header` property in sapum.properties [Page 102].

Anonymous Logon

Use

Anonymous logon allows users to access the portal in anonymous mode, without providing any form of authentication. For example, if your company sets up an external portal that is accessible through the Internet, you can make anonymous content available to anyone who wants to visit the portal. Using self-registration, visitors can then register themselves as portal users.

Restrictions

Currently SAP Knowledge Management objects do not support anonymous logon.

Integration

Modes of Anonymous Logon

SAP Enterprise Portal provides two forms of anonymous logon:

- **Anonymous logon with named anonymous users (default configuration)**

This form of anonymous logon uses ‘named’ anonymous users, which are users that exist either in the user data store or as service users. These users are automatically assigned to the group *Anonymous Users*. You can assign roles containing anonymous content to the users individually or to the group *Anonymous Users*.

- **Simple anonymous logon**

  With this form of anonymous logon there is no physical user in the data store, so, for example, you cannot assign a role containing anonymous content to an anonymous user. As there is no user, a logon ticket is **not** issued.

  If you use simple anonymous logon, there is no current user. This means that personalization functions such as modifying the user’s profile are not available.

For anonymous logon the following properties in `sapum.properties` are relevant:

Property	Value	Description
`ume.logon.anonymous_user.mode`	`1` = Anonymous logon with named anonymous users is used. (Default value) `0` = Simple anonymous logon is used.	Defines which mode of anonymous logon is to be used.
`ume.login.guest_user.uniqueids`	Comma-separated list of user IDs. The default value is `anonymous`.	Only required if `ume.logon.anonymous_user.mode=1.` Defines which users are the named anonymous users. These users automatically belong to the default group *Anonymous Users*.  The administrator has to create these anonymous users in the user data store
`ume.login.guest_user.defaultid` (Optional)	`<no_value>` = The first user in the list for `ume.login.guest_user.uniqueids` is used. `<User ID>`	Defines which anonymous user is used for anonymous logon if the parameter `j_user` in the portal URL is empty.

For more information on how to set these properties, see *Enterprise Portal Administration Guide → Portal Platform → System Administration → User Management Configuration.*

In the following excerpt from `sapum.properties`, the users `anon1`, `anon2`, and `anon3` are defined as anonymous users and, if no user is specified in the portal URL, `anon2` is used for anonymous access to the portal.

```
##################################################################
######
    anonymous user
##################################################################
######
```

```
ume.login.anonymous_user.mode=1
ume.login.guest_user.uniqueids=anon1,anon2,anon3
ume.login.guest_user.defaultid=anon2
```

Authentication Schemes for Anonymous Logon

The following authentication schemes that are shipped with SAP Enterprise Portal support anonymous logon:

- `guest`
- `anonymous`

See also Authentication Schemes Shipped with SAP Enterprise Portal [Page 16].

Features

The following table illustrates the features that are available depending on which authentication mode you use with which authentication scheme.

	Anonymous logon with named anonymous users (`ume.login.anonymous_user.mode=1`)	**Simple anonymous logon** (`ume.login.anonymous_user.mode=0`)
Authentication Scheme = guest	Named anonymous users. SAP logon ticket is issued.	No named anonymous users. SAP logon ticket is **not** issued.
Authentication Scheme = anonymous	Named anonymous users. SAP logon ticket is **not** issued.	No named anonymous users. SAP logon ticket is **not** issued.

Activities

You can define anonymous logon at iView level or at portal level.

- For an example of setting up the **complete portal** for anonymous logon with named anonymous users, see Configuring Anonymous Logon with Named Anonymous Users [Page 27].
- Alternatively you can define an **individual iView** as anonymous content by setting the value of the iView parameter *Authentication Scheme* to `guest` or `anonymous`. See Assigning an Authentication Scheme to an iView [Page 14]. Users can launch an anonymous iView using the direct URL for that iView without having to provide authentication.

  For example, users can call up the self-registration iView directly using the following URL:

  ```
  http://<server>:<port>/irj/servlet/prt/portal/prtroot/usermanagementadmin.SelfReg
  ```

Configuring Anonymous Logon with Named Anonymous Users

Use

This procedure describes how to configure the portal for anonymous logon with named anonymous users and using `anonymous` as the authentication scheme. In this case anonymous users are not issued with a SAP logon ticket.

By setting up anonymous logon with one or more named anonymous users, you can assign roles containing anonymous content to the named anonymous users. You can either assign the roles to the users individually or to the group *Anonymous Users*. If you define more than one anonymous user, you can assign different roles to the different anonymous users and set up different URLs to the portal, allowing you to control the anonymous content that portal users see.

Prerequisites

Check that the user management properties are correctly set

To set up the portal for anonymous logon, the user management properties should be set as follows:

```
ume.logon.anonymous_user.mode=1
ume.login.guest_user.uniqueids=<list_of_anonymous_users>
```

```
ume.logon.anonymous_user.mode=1
ume.login.guest_user.uniqueids=anon1,anon2,anon3
```

Procedure

Create named anonymous users

1. Create the anonymous users that you defined in `ume.login.guest_user.uniqueids`.

   For example, create users with the user IDs `anon1`, `anon2`, and `anon3`.

   After you restart the Java application server, these users are automatically in the *Anonymous Users* group.

Create anonymous content

2. Create a role in which, for all pages and iViews, *Authentication Scheme* is set to `anonymous`.

   For more information on creating roles, see *SAP Enterprise Portal Administration Guide → Portal Platform → Content Administration.*

   As this role will be available to anonymous users, you should ensure that it does not contain sensitive content, for example administration functions.

Assign anonymous content to anonymous users

3. Assign the anonymous role you created to one of the anonymous users or to the *Anonymous Users* group.

Create a copy of PortalLauncher iView and set its authentication scheme to anonymous

4. Choose *Content Administration* → *Portal Content*.

   The Portal Content Studio appears.

5. In the Portal Content Studio, choose *New from Portal Archive* → *iView*.

6. In the iView Wizard, choose *com.sap.portal.navigation.portallauncher* and continue through the wizard.

   When the wizard is completed, the Property Editor is displayed.

7. In the Property Editor, change the *Authentication Scheme* to `anonymous`.

8. Save your changes.

9. Make a note of the ID of your anonymous PortalLauncher iView.

   For example:
   `pcd:portal_content/myFolder/iViews/com.sap.AnonPortallauncher`

Redirect the portal to the copy of the PortalLAuncher iView

When a user starts the portal using the URL `<server>:<port>/irj`, the URL is redirected to the PortalLauncher iView. You have to change the redirect so that the portal is redirected to the copy of PortalLauncher you just created.

10. On the file server, open the following file:

    ```
    SAP_J2EEngine6.20\cluster\server\services\servlet_jsp\work\jspTemp\irj\root\index.html
    ```

    This file contains the following line:

    ```
    <body
    onload="location.replace('servlet/prt/portal/prtroot/com.sap.portal.navigation.portallauncher.default' +
    document.location.search)"></body>
    ```

11. Change this line to the following:

    ```
    <body
    onload="location.replace('servlet/prt/portal/prtroot/<ID_of_anonymous_PortalLauncher>' + document.location.search)"></body>
    ```

    For example, if the ID of your anonymous PortalLauncher iView is `pcd:portal_content/myFolder/iViews/com.sap.AnonPortallauncher`, change the line to the following:

    ```
    <body
    onload="location.replace('servlet/prt/portal/prtroot/pcd!3aportal_content!2fmyFolder!2fiViews!2fcom.sap.AnonPortallauncher' + document.location.search)"></body>
    ```

12. Restart the java application server.

Result

When users launch the portal, they are logged on as the first user in the list of anonymous users (in this example, `anon1`). They do not have to provide any form of authentication, unless one of the pages or iViews in the role assigned to the anonymous user is not set to `anonymous`. In this case, a logon screen appears in the page or iView.

A log on link appears in the header area. When the user clicks on this link, the form-based logon screen appears giving users the option to register as portal users.

Possible Variations

Use a specific named anonymous user for anonymous logon

The URLs to access the portal can optionally contain a `j_user` parameter that specifies the user to be used for anonymous logon.

In the following example, the portal is accessed with the anonymous user `anon2`:

```
http://<server>:<port>/irj/index.html?j_user=anon2
```

Issue SAP logon tickets to named anonymous users

If you wish named anonymous users to be issued with SAP logon tickets when they access the portal, use the `guest` authentication scheme instead of `anonymous` when defining your anonymous content.

Issuing SAP logon tickets to anonymous users can have an impact on the security of your portal. If a user account with the same user ID as the named anonymous user exists in any system that is accessed via Single Sign-On with logon tickets through the portal (for example, an SAP R/3 System), the guest user can access this system.

Customizing the Logon Screens

Use

The portal is shipped with a standard set of logon screens. These include the screen in which users enter their user ID and password, the screen for requesting help from an administrator, and so on. If required, you can change these screens to reflect your company's look and feel. To do this, you have to modify the Java Server Pages (JSPs) of the logon component shipped by SAP and reconfigure user management to use the modified logon component.

The standard logon component containing the code and resources used by the logon screens is shipped in a portal archive (PAR) file named `com.sap.portal.runtime.logon.par`.

Procedure

Make a Copy of the Standard Logon Component and Modify It

1. Navigate to `<SAP_J2EE_Engine_installation_dir>\cluster\server\services\servlet_jsp\work\jspTemp\irj\root\WEB-INF\deployment\pcd`.

   There you can find `com.sap.portal.runtime.logon.par.bak`.

2. Make a copy of `com.sap.portal.runtime.logon.par.bak` and rename it. In this example, we renamed it to `my.new.logon.par`.

3. Move `my.new.logon.par` to a location outside of `<SAP_J2EE_Engine_installation_dir>`.

4. Extract all files from `my.new.logon.par` preserving the directory structure.

5. Modify the files in the extracted PAR file as required.

6. Put the modified files back into `my.new.logon.par`.

7. Copy `my.new.logon.par` back to `<SAP_J2EE_Engine_installation_dir>\cluster\server\services\servlet_jsp\work\jspTemp\irj\root\WEB-INF\deployment\pcd`.

Configure User Management to Use the New Logon Component

8. Navigate to `<SAP_J2EE_Engine_installation_dir>\cluster\server\ume`.

   There you can find `authschemes.xml.bak`.

9. Rename `authschemes.xml.bak` to `authschemes.xml`.
10. Open `authschemes.xml` and replace all occurrences of the string "`com.sap.portal.runtime.logon`" with "`my.new.logon`" in the tags `<frontendtarget>`.
11. Save the file.
12. Restart the portal.

Result

The modified logon screens are displayed at logon.

Authorizations

Authorizations define which objects users can access and which actions they can perform. The portal has an authorization concept that is implemented using permissions, security zones, and the *AuthRequirement* property. These are described in more detail below.

- **Permissions**: permissions for all Portal Content Directory (PCD) objects. Portal permissions define portal user access rights to portal objects in the PCD and are based on access control list (ACL) methodology. Essentially, every portal object can be assigned directly to an individual user or collectively to groups of users through user groups and roles. Portal content objects for which you can set permissions are folders (Portal Catalog folders, not role folders), iViews, pages, layouts, roles, worksets, packages, and systems. When any portal user accesses a portal tool that displays portal objects stored in the PCD, those objects are filtered according to the user's access permissions. If a user is permitted to access a portal object, the permission level set for the user defines which actions and operations the user can perform on that object. Permissions also define which objects are available to end users in a runtime portal environment.

  For more information on permissions, see *SAP Enterprise Portal Administration Guide → Portal Platform → System Administration → Portal Permissions.*

- **Security Zones**: permissions for portal components. A means of implementing an additional layer of security to portal components and services which are accessed by a URL. Access is controlled by means of progressive safety levels and portal permissions which are assigned to authorized users. Security zones are defined in portal components at the development phase. Portal applications that are not assigned to a security zone can only be accessed via an iView, not a direct URL. Therefore, it is only necessary to define a security zone for portal applications that are not launched via iViews. Information on defining security zones at the code level in portal components is provided in detail in the Portal Development Kit (PDK) documentation, which is available on the iViewStudio at www.iviewstudio.com.

  For more information on security zones, see *SAP Enterprise Portal Administration Guide → Portal Platform → System Administration → Portal Permissions → Security Zones.*

- **AuthRequirement property**: This is a master iView property used in EP 5.0 that defines which users are authorized to access a master iView or Java iViews derived from a master iView. For backward compatibility with iViews developed for EP 5.0, EP 6.0 supports this property.

  For details on the *AuthRequirement* property, see *SAP Enterprise Portal 5.0 Administration Guide → iViews → Master iViews → Master iView Properties → Portal Component Properties*.

In the portal, roles are only indirectly linked to authorization. Portal roles group together the portal content required by users with a certain role in the company. In addition, the role structure defines the navigation structure that a user sees in the portal. Users and groups assigned to a role inherit the permissions of the role. By default this is end user permission.

Single Sign-On

Single Sign-On (SSO) is a key feature of the Enterprise Portal that eases user interaction with the many component systems available to the user in a portal environment. Once the user is authenticated to the enterprise portal, he or she can use the portal to access external applications. With SSO in the Enterprise Portal, the user can access different systems and applications without having to repeatedly enter his or her user information for authentication.

The Enterprise Portal SSO mechanism is available in two variants depending on security requirements and the supported external applications:

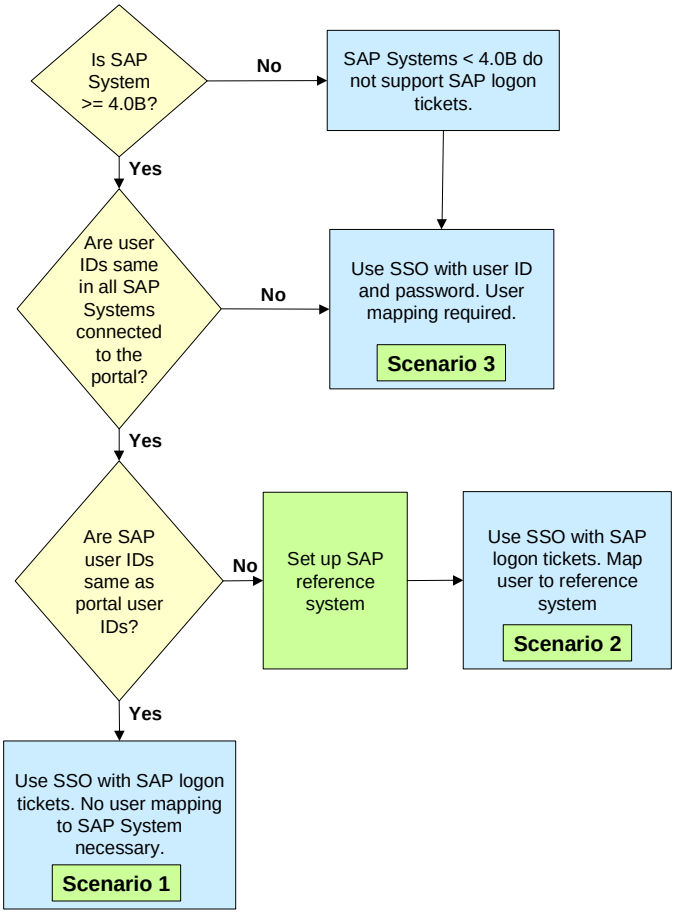
- SSO with SAP logon tickets
- SSO with user ID and password

Both variants eliminate the need for repeated logons to individual applications after the initial authentication at the enterprise portal. Whereas SSO with SAP logon tickets is based on a secure ticketing mechanism, SSO with user ID and password forwards the user's logon data (user ID and password) to the systems that a user wants to call.

Single Sign-On to SAP Systems

This section summarizes the different scenarios for Single Sign-On to SAP Systems. Which method of Single Sign-On (SSO) you use with a SAP System depends on various parameters, such as the release of the system. There are different prerequisites, for example, users must have the same user ID in all SAP Systems that are accessed via SSO with SAP logon tickets.

The following diagram helps you find out which method of Single Sign-On to use with a specific SAP System.



Scenario 1: Single Sign-On using SAP logon tickets without user mapping

Users must have the same user IDs in all SAP systems that are accessed via SSO with SAP logon tickets. If the SAP user IDs are the same as the portal user IDs, user mapping is not required. You need to perform the following steps:

1. Configure Portal Server for SSO with SAP Logon Tickets [Page 35]
2. Configure SAP Systems to Accept and Verify SAP Logon Tickets [Page 37]

Scenario 2: Single Sign-On using SAP logon tickets with user mapping

If users have different users IDs in the SAP Systems than in the portal, you must define a SAP reference system and map each user's user ID to their user ID in the reference system. You must perform the following steps:

1. Define an SAP R/3 Reference System for User Data [Page 34]
2. Configure Portal Server for SSO with SAP Logon Tickets [Page 35]
3. Configure SAP Systems to Accept and Verify SAP Logon Tickets [Page 37]
4. Each user must map his or her user ID to his or her user ID in the SAP Reference System as described in *Enterprise Portal Administration Guide → Portal → User Administration → User Mapping → Mapping Users: User Enters Own Data*.

Scenario 3: Single Sign-On using user ID and password with user mapping

There are two cases where you would use this method of Single Sign-On:

- The SAP System has release 3.1I.
- Users have a different user ID in the SAP System in question than in the reference SAP System used for logon tickets.

You must perform the following step: Configuring SSO with User ID and Password to SAP Systems [Page 45].

Defining an SAP Reference System for User Data

Use

When you use SAP logon tickets for Single Sign-On to SAP Systems, users must have the same user IDs in all SAP Systems that are configured to use SAP logon tickets. If the SAP user IDs are different to the portal user IDs, you must define an SAP reference system. Users then map their portal user ID to the user ID in the SAP reference system.

The mapped user ID is included in the SAP logon ticket and enables Single Sign-On using logon tickets to all SAP Systems in which the user has the same user ID.

Prerequisites

Users have the same ID in all SAP component systems that are configured to use logon tickets for Single Sign-On. Passwords do not have to be identical.

Procedure

Define a system object for the reference system

1. If the system you wish to use as SAP reference system has not been defined as a system yet in the portal, define it as described in *Enterprise Portal Administration Guide → Portal → System Administration → System Landscape → System Landscape Editor → Creating a System Landscape Object*.
2. Ensure that a system alias has been defined for the system. If it does not have a system alias, it will not appear in the user mapping tool.
3. In the system's properties, set the property *R/3 reference system* to *1*.
4. If required, also set the user mapping properties. For details, see *Enterprise Portal Administration Guide → Portal Platform → User Administration → User Mapping → System Properties for User Mapping*.
5. Save your changes.

Define the reference system in the user management configuration tool

6. In the user management configuration tool, choose *Security Settings*.

   For more information on the user management configuration tool, see *Enterprise Portal Administration Guide → Portal Platform → System Administration → User Management Configuration → User Management Configuration Tool.*.
7. In *R/3 Reference System*, enter the system alias of the above system.
8. Restart the Java application server.

Result

When users start the user mapping function, one of the component systems that they can select is the SAP reference system. They can map their portal user ID to their user ID in this reference system. The user mapping function connects to the SAP reference system using the user ID and password to verify that the password entered by the user is correct.

The next time the user logs on to the portal, the portal generates an SAP logon ticket for the user that contains both his or her portal user ID and mapped user ID.

Single Sign-On with SAP Logon Tickets

Purpose

SAP logon tickets represent the user credentials. The Portal Server issues a logon ticket to a user after successful initial authentication. The logon ticket itself is stored as a cookie on the client and is sent with each request of that client. It can then be used by external applications such as SAP systems to authenticate the portal user to those external applications without any further user logons being required.

SAP logon tickets contain information about the authenticated user. They do not contain any passwords. Specifically, logon tickets contain the following items:

- Portal user ID and one mapped user ID for external applications
- Authentication scheme
- Validity period
- Information identifying the issuing system
- Digital signature

Technically, SSO with SAP logon tickets works as follows:

1. The first time the Portal Server is started, it generates a cryptographic key pair. The private part of this key is used for ticket generation (for the digital signature).
2. Once the user has been successfully authenticated in the portal, the Portal Server issues a logon ticket to the user. This logon ticket is stored as a non-persistent cookie in the browser on the client.
3. Each time the user tries to access an external system from the portal, the Portal Server sends the logon ticket with the request to the external system.
4. The external system checks that the logon ticket is valid by verifying the digital signature of the Portal Server. It uses the public key contained in the digital certificate of the Portal Server to verify this.
5. If the logon ticket is valid, the external system extracts the user ID for that system from the logon ticket.
6. The user is logged on to the external system without having to enter his or her user ID and password.

The Portal Server issues a SAP logon ticket for the Internet domain or a sub-domain of the Portal Server only.

Process Flow

To allow Single Sign-On using SAP logon tickets between the portal and its component systems you must perform the following steps:

1. Configure the Portal Server to allow Single Sign-On with SAP logon tickets. This step is optional, as by default the portal is configured for SAP logon tickets.
2. Configure the component systems to accept and verify SAP logon tickets.

Configuring Portal Server for SSO with SAP Logon Tickets

Use

In the default mode, the Portal Server creates and digitally signs SAP logon tickets for users, therefore you do not need to make any settings. However there are some settings that you need to make in particular cases. These are described below.

Procedure

Configure the lifetime of the SAP logon ticket

You set the lifetime of the SAP logon ticket in the user management configuration tool. For details, see *Enterprise Portal Administration Guide → Portal → System Administration → User Management Configuration → User Management Configuration Tool*.

Map portal user IDs to user IDs in other systems

If users' portal user IDs are different to their user IDs in the component systems, the administrator or user must map the portal user ID to the user ID in the other systems. For details, see *Enterprise Portal Administration Guide → Portal → User Administration → User Mapping*.

If you have several SAP component systems in your portal landscape, and the SAP users have not been synchronized with the portal users, you define a reference system for user data and map the portal users to the users in this system. For more information, see Defining an SAP R/3 Reference System for User Data [Page 34].

SAP Systems only: Set logon method to SAP logon tickets in portal system landscape

For each SAP System that you wish to access with SAP logon tickets, do the following:

1. Open the system for property editing as described in *Enterprise Portal Administration Guide → Portal → System Administration → System Landscape → System Landscape Editor → Editing System Properties*.
2. Set the value of the property *Logon Method* to *SAPLOGONTICKET*.
3. Save your changes.

Configuring Component Systems for SSO with SAP Logon Tickets

When a user calls an external application, his or her logon ticket is passed on to the appropriate application or information system where it is checked to see if it is valid. In order to work with SAP logon tickets, the external application has to perform three tasks as follows:

1. The external system has to make sure that a trusted Portal Server has issued the ticket.
2. The digital signature in the ticket of the Portal Server needs to be verified. The first two steps require the digital certificate of the issuing Portal Server.
3. If the ticket is valid, the appropriate user ID contained in it has to be extracted.

This verification procedure is standard in SAP systems. For information on how to configure SAP Systems, see Configuring SAP Systems to Accept and Verify SAP Logon Tickets [Page 37].

Configuring SAP Systems to Accept and Verify SAP Logon Tickets

Use

The Portal Server digitally signs SAP logon tickets as it issues them to the portal users. SAP Systems need to accept the tickets and verify the Portal Server's digital signature. The following information is important for the SAP System to be able to accept and verify SAP logon tickets:

- The SAP System should only accept SAP logon tickets issued from their designated Portal Server. Therefore, the identity of the Portal Server needs to be entered in the SAP System's SSO access control list (ACL).
- The SAP System needs to be able to verify the Portal Server's digital signature. The Portal Server has a self-signed certificate, therefore the SAP System needs access to the Portal Server's public-key information, which needs to be entered in the SAP System's certificate list.

Prerequisites

- The SAP System has Release 4.0B or higher. SAP logon tickets are not supported in releases lower than 4.0B.
- The Enterprise Portal Plug-In that corresponds to the Enterprise Portal release has been installed in the SAP System.
- The required kernel patches have been applied to R/3 Systems prior to Release 4.6C. For more information, see the section on implementing new kernels for the SAP Application Server in SAP Note 177895. Note that after applying the kernel patches, you may need to patch the operating system of the R/3 System so that the new kernel works.
- Users must have the same user IDs in all SAP Systems that are accessed via Single Sign-On with SAP logon tickets. If the SAP user IDs are different to the portal user IDs, you must define a SAP reference system. See Defining an SAP R/3 Reference System for User Data [Page 34].
- The SAP Security Library is installed on all of the system's application servers. For best practices, we recommend installing the most recent version of the library, which is available on the `sapserv<x>` under `/general/misc/security/SAPSECU/<platform>`.
- You have configured the Portal Server for Single Sign-On with logon tickets. See Configuring Portal Server for SSO with SAP Logon Tickets [Page 35].

Procedure

In SAP systems with Release 4.6C or higher you can use transaction STRUSTSSO2 to complete the first 2 steps of the following procedure. This is described in Using Transaction STRUSTSSO2 in SAP System >= 4.6C [Page 39].

Add Portal Server to ACL of component system

The Portal Server is identified by system ID, client, and the name in the certificate. You must enter these details in the access control list of the component system as follows.

1. In the component system, maintain table *TWPSSO2ACL* with transaction *SM30*.
2. Create a new entry for the Portal Server by choosing *New entries*.

3. Enter the portal's system ID and client. By default, the portal's system ID is the common name (CN) of the Distinguished Name entered during installation of the portal. The default client is 000.

   If necessary, you can change these default values by changing the properties `login.ticket_issuer` and `login.ticket_client` respectively in user management properties.

4. Enter the following values for *Subject name*, *Issuer name*, and *Serial number*.

Field	Value
Subject name	Distinguished name (DN) of owner of portal server certificate. This is the DN that was entered during installation of the portal. For example: `CN=EP6, OU=Portal Installation, OU=Enterprise Portal, O=SAP Trust Community, C=DE`
Issuer name	Distinguished name of issuer of portal server certificate. If the portal is using a self-signed certificate, this is the same as the above entry.
Serial number	00

You can look up the subject name, issuer name, and serial number of the portal server certificate in the Keystore Manager [Page 46].

5. Save your entries.

Import public-key certificate of Portal Server to component system's certificate list

This procedure is release-specific.

- If the SAP component system is based on Release 4.6C or higher, follow the procedure detailed in Importing Portal Certificate into SAP System >= 4.6C [Page 40].
- If the SAP component system is based on Release 4.0B to 4.6B, follow the procedure detailed in Importing Portal Certificate into SAP System < 4.6C [Page 42]

Set profile parameters

On all of the component system's application servers:

1. Set the profile parameters `login/accept_sso2_ticket` = 1 and `login/create_sso2_ticket` = 0 in every instance profile.
2. For Releases 4.0 and 4.5, also set the profile parameter `SAPSECULIB` to the location (path and file name) of the SAP Security Library.

Set ITS service parameters

On each of the ITS servers of the SAP component system, in the global service file `global.srvc` , set the following parameters:

Set the Parameter	To the Value	Comment
`~login`	(space)	
`~password`	(space)	

`~mysapcomusesso2cookie`	1	Enables the user to log on to the system using an existing SAP logon ticket.

Result

The SAP component systems are able to accept SAP logon tickets and verify the Portal Server's digital signature when they receive a logon ticket from a user.

Using Transaction STRUSTSSO2 in SAP System >= 4.6C

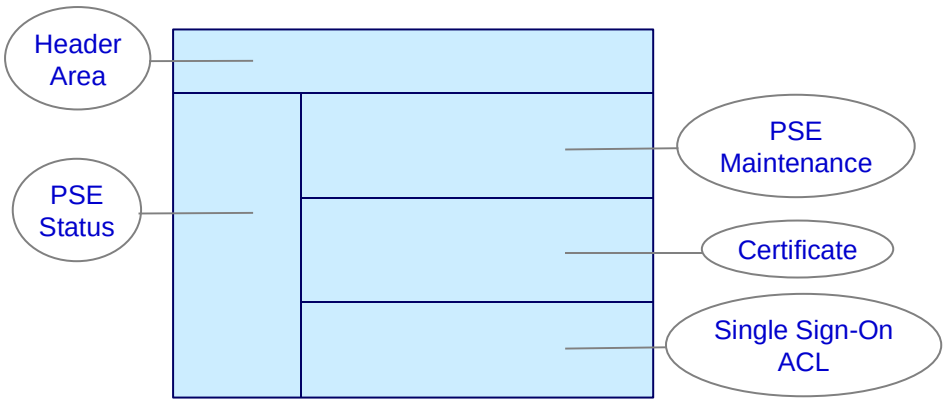
Procedure

Import public-key certificate of Portal Server to component system's certificate list and add Portal Server to ACL of component system

Both of these steps can be performed with transaction *STRUSTSSO2*, which is an extended version of transaction *STRUST*. For detailed documentation on transaction *STRUST*, see the Web Application Server documentation under *Security* → *Trust Manager.*

1. In the SAP System, start transaction *STRUSTSSO2*.

   A screen with the following layout appears.



   The **PSE status** frame on the left displays the PSEs that are defined for the system.

   The **PSE maintenance** section on the top right displays the PSE information for the PSE selected in the PSE status frame.

   Below that, the **certificate** section displays certificate information for a certificate that you have selected or imported.

The **Single Sign-On ACL** section on the bottom right displays the entries in the ACL of the system.

Note that the layout of the transaction will vary slightly, depending on the release of the SAP System.

2. In the PSE status frame on the left, choose the *system* PSE.
3. In the certificate section, choose *Import Certificate*.

   The *Import Certificate* screen appears.
4. Choose the *File* tab.
5. In the *File path* field, enter the path of the portal's verify.der [Page 102] file.
6. Set the file format to *DER coded* and confirm.
7. In the Trust Manager, choose *Add to PSE*.
8. Choose *Add to ACL*, to add the Portal Server to the ACL list.
9. In the dialog box that appears, enter the portal's system ID and client. By default, the portal's system ID is the common name (CN) of the Distinguished Name entered during installation of the portal. The default client is 000. If necessary, you can change these default values by changing the properties `login.ticket_issuer` and `login.ticket_client` respectively in user management properties.

   The other values are taken from the certificate.
10. Save your entry.
11. Do not forget to set profile parameters and ITS service parameters as described in Configuring SAP Systems to Accept and Verify SAP Logon Tickets [Page 37].

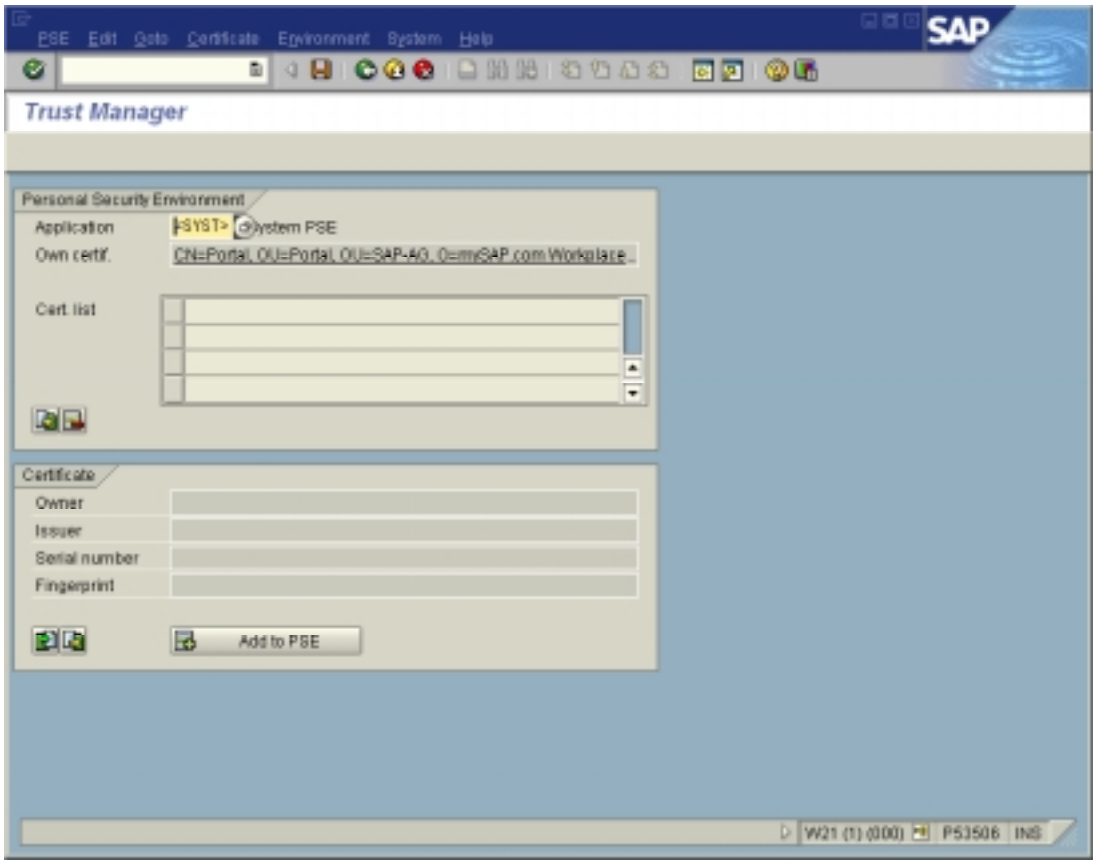
Result

The SAP component systems are able to accept SAP logon tickets and verify the Portal Server's digital signature when they receive a logon ticket from a user.

Importing Portal Certificate into SAP System >= 4.6C

1. In the component system, start transaction STRUST.

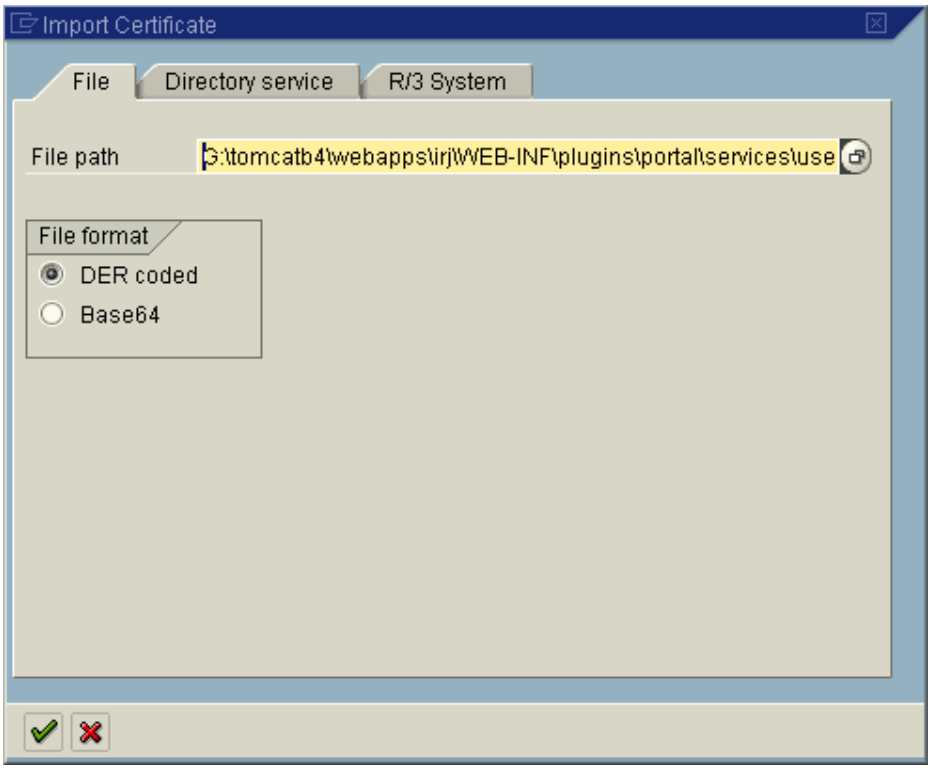
   The following screen appears.



   This screen displays a list of the certificates contained in the PSE of the component system.

2. In the certificate group box, choose *Import Certificate*.

   The *Import Certificate* screen appears.



3. Choose the *File* tab.
4. In the *File path* field, enter the path of the portal's verify.der [Page 102] file.
5. Set the file format to DER coded and confirm.
6. In the Trust Manager, choose *Add to PSE*.
7. Save the new certificate list.

   The new certificate list is automatically replicated to all application servers in the system. You do not have to import the portal certificate onto each application server separately.

Importing Portal Certificate into SAP System < 4.6C

Check whether there is a file `DIR_PROFILE\SAPSSO2.pse` in the profile directory of the component system. (*DIR_PROFILE* is a profile parameter).

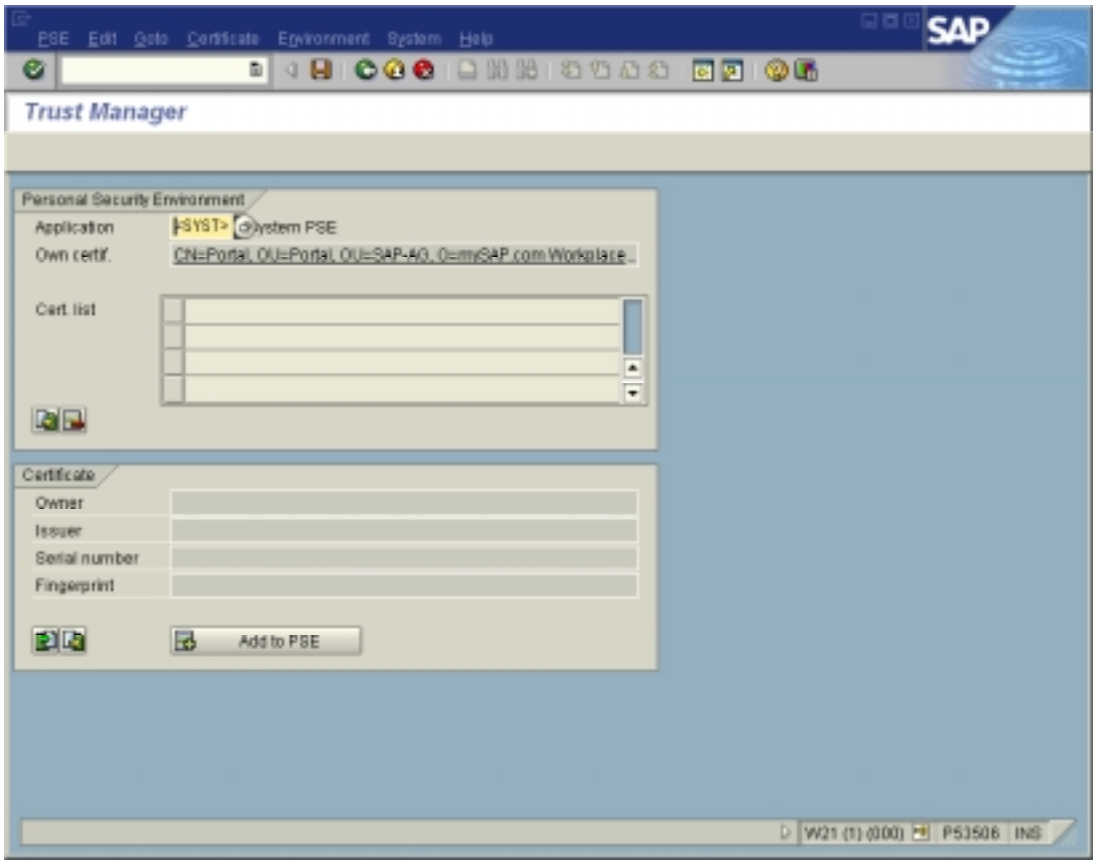
- If not, copy the verify.pse [Page 102] file from the portal into the *DIR_PROFILE* directory of the component system and rename it to `SAPSSO2.pse`.
- If yes, check whether it is still needed, for example if there is a current SAP Workplace installation that will still be used after SAP Enterprise Portal is set up. If it is not needed, replace it with the renamed verify.pse [Page 102]. If it is still needed, perform the steps outlined below.

If you need to keep the existing PSE file in your component system

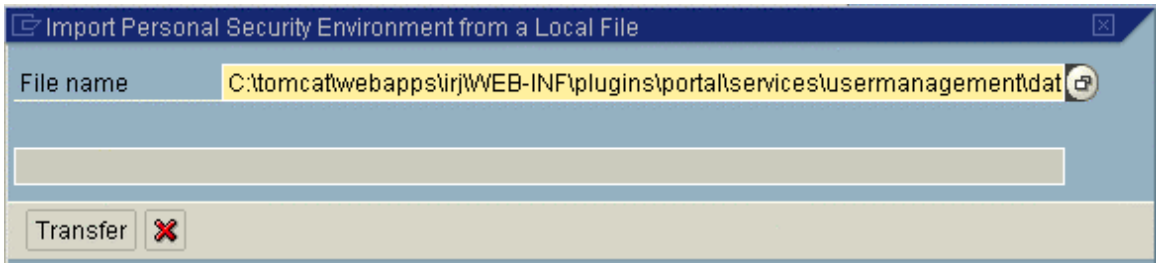
You can use the Trust Manager to import the portal certificate into the existing PSE file. To do this, you need any SAP System with Release 4.6C or higher.

1. In a SAP System with Release 4.6C or higher, start transaction *STRUST*.

   The following screen appears.



2. In the *Application* field, choose <*FILE*>, and enter the path to the `SAPSS02.pse` file of your component system. Choose *Transfer*.



3. Choose *Import certificate*. Enter `verify.der` from Portal Server.
4. Choose *Add to PSE*.
5. Save the updated `SAPSS02.pse`.
6. Copy the updated `SAPSS02.pse` to the *DIR_PROFILE* directory of your component system.

Using More Than One Portal

Use

In some cases you may want to allow two portals to access the same R/3 component system via Single Sign-On with SAP logon tickets.

Each portal installation is uniquely identified by system ID, client, and the distinguished name in the portal server certificate. If you want to connect two portals to the same R/3 system, the combination of these three items must be unique for each portal, so that the R/3 System can tell them apart.

The following table provides an overview of distinguished name, system ID, and client. If these values are the same for both portal installations, you will need to change one of the values on one of the portal installations.

	Default value	**How can I change it?**
Distinguished name	Distinguished name entered during installation of the portal.	Create a new portal server certificate (and cryptographic key pair) using the Keystore Manager [Page 46].  If you create a new certificate for a portal installation, you will have to reconfigure Single Sign-On for all backend systems that accept SAP logon tickets.
System ID	Common name (CN) of Distinguished name entered during installation.	Change the value of the `login.ticket_issuer` property in user management properties.
Client	000	Change the value of the `login.ticket_client` property in user management properties.

Single Sign-On with User ID and Password

Purpose

The Single Sign-On (SSO) mechanism with user name and password provides an alternative for applications that cannot accept and verify SAP logon tickets. With this SSO mechanism the Portal Server uses user mapping information provided by users or administrators to give the portal user access to external systems. The portal components connect to the external system with the user’s credentials.

As the user's user ID and password are sent across the network, you should use a secure protocol such as Secure Sockets Layer (SSL) for sending data.

Process Flow

There are different procedures depending on the requirements.

Single Sign-On to SAP Systems

You can access SAP Systems that do not support SAP logon tickets via Single Sign-On with user ID and password. These are SAP Systems with release 3.1I. For more information, see Configuring SSO with User ID and Password to SAP Systems [Page 45].

Single Sign-On to non-SAP systems via a Java iView developed specifically for the customer

The system must be defined in the system landscape. For details, see *Enterprise Portal Administration Guide → Portal → System Administration → System Landscape → System Landscape Editor → Creating a System Object*.

The administrator or user must map user data to user data in the system. For more information, see *Enterprise Portal Administration Guide → Portal → User Administration → User Mapping*.

The iView through which the user tries to access the system must be programmed to get the mapped user data from the data repository and write the user credentials (user ID and password) in a header field of the request. The system can then log on the user with these credentials. This can be done using the Java APIs provided with SAP Enterprise Portal.

Configuring SSO with User ID and Password to SAP Systems

Use

This procedure describes how to configure SAP Enterprise Portal and a SAP System for Single Sign-On with user mapping. In general we recommend using Single Sign-On with SAP logon tickets or client certificates. Single Sign-On with user ID and password should only be used if no other Single Sign-On method is possible. It has the following advantages:

- It can be used for Single Sign-On to SAP Systems that do not support SAP logon tickets (that have a release lower than 4.0B).
- You do not have to have Central User Administration (CUA) in place. Users can have a different user ID and password in the SAP System in question than in the reference SAP System used for the logon ticket.

  When Single Sign-On with user ID and password is used, the user ID and password are transmitted in plain text using HTTP POST. We strongly recommend that you protect the connections to the SAP System using HTTPS or SNC to prevent the user ID and password being eavesdropped by an external party.

Procedure

1. In the system object defining the SAP System in the portal, set the property *Logon Method* to *UIDPW*. For more information on defining system objects, see *Enterprise Portal Administration Guide → Portal Platform → System Administration → System Landscape*.
2. Either the administrator or the users must map users' user ID and password to their user ID and password in the SAP System. For more information on user mapping, see *Enterprise Portal Administration Guide → Portal Platform → User Administration → User Mapping.*

Result

When the user tries to access the SAP System through the portal, the user mapping information is used to access the component system

 Keystore Manager

Use

The keystore manager is a tool that allows you as administrator to manage the keystore `ticketKeyStore`. `ticketKeyStore` contains the following security information:

- The portal server's public and private key
- The portal server's server certificate
- All the certificates of Certification Authorities (CAs) that are trusted by the portal server
- Additional certificates of entities trusted by the portal server

`ticketKeyStore` can only contain **one** private key.

Integration

The keystore manager is based on the portal component *com.sap.portal.usermanagement.admin.KeystoreComponent*. It is currently not included in any of the administration roles shipped with the portal.

Features

With the keystore manager you can:

- View contents of `ticketKeyStore`
- Create a new `ticketKeyStore` for the portal with a new key pair and a new distinguished name (DN)
- Import certificates into `ticketKeyStore`
- Delete certificates from `ticketKeyStore`
- Generate server certificate requests
- Import responses to certificate requests into `ticketKeyStore`

Activities

Accessing the Keystore Manager

To access the keystore manager, you have the following options:

- Access it using the following direct URL:

  ```
  <portal_host>:<port>/irj/servlet/prt/portal/prtroot/
  com.sap.portal.usermanagement.admin.KeystoreComponent
  ```

- Access it using the PortalAnywhere utility:

  a. Call up the following URL:

  ```
  <portal_host>:<port>/irj/servlet/prt/portal/prtroot/PortalA
  nywhere.Go
  ```

  b. Navigate to *com.sap.portal.usermanagement.admin* → *KeystoreComponent and choose Go.*

Using the Keystore Manager

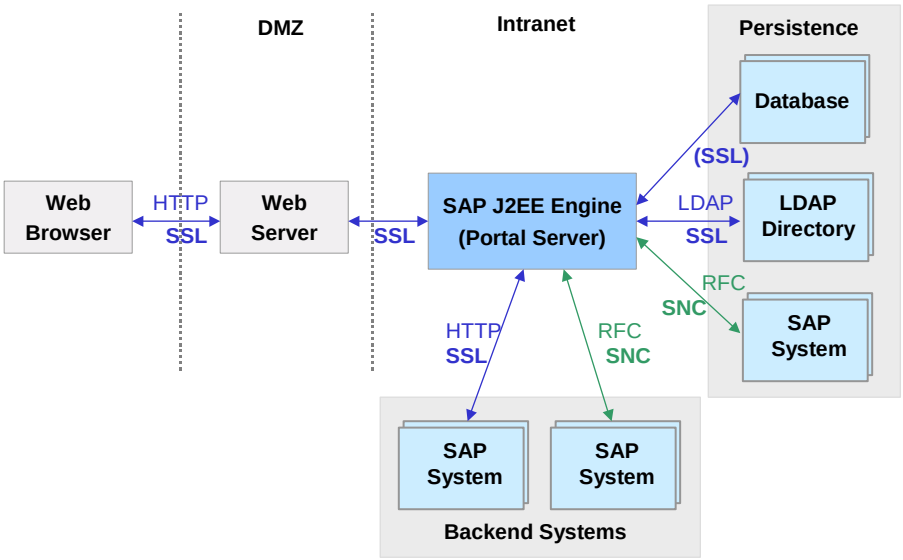
Activity	Action
View contents of `ticketKeyStore`	Choose *Content*.
Delete certificates from `ticketKeyStore`	1. Choose *Content*. 2. Select the certificate you wish to delete and choose *Delete*.
Import certificates of trusted entities into `ticketKeyStore`	1. Choose *Import Trusted Certificate*. 2. Browse to the certificate file. 3. Enter an alias for the certificate. 4. Choose *Upload*.
Create a server certificate request	Choose *PKCS#10* → *Create PKCS#10 Request*. You can copy the resulting certificate request from the text area and send it to a certification authority (CA) to request a server certificate.
Import responses to certificate requests into `ticketKeyStore`	1. Choose *PKCS#10*. 2. Paste the response from the CA into the text area. 3. Choose *Import CA Response*.
Create a new `ticketKeyStore` for the portal with a new key pair and a new distinguished name (DN)  If you create a new `ticketKeyStore`, you have to reconfigure Single Sign-On to backend systems that accept the SAP logon ticket	1. Choose *Create Keystore*. 2. Enter a distinguished name (DN) for the portal server certificate. For each attribute of the DN, choose it from the drop-down list, enter a value in the corresponding field, and choose *Add*. 3. Choose *Create as Ticket Keystore*.

Secure Communications

Protecting the information transferred between the client and the Portal Server and between the internal components of the SAP Enterprise Portal is important. The data transferred contains authentication credentials and possibly other sensitive data that must not be known to third parties. This kind of data must be encrypted using secure communication protocols such as Secure Sockets Layer (SSL) or Secure Network Communications (SNC).

We recommend that you protect all communication channels used during normal operation of the SAP Enterprise Portal.

The following diagram provides an overview of the communication channels between the components of the Enterprise Portal.



This diagram displays a secure network architecture where a Web server is placed in a demilitarized zone (DMZ) in front of the Portal Server. It is also possible to have a network architecture in which the client communicates directly with the Portal Server.

The Portal Server uses a database to store portal-related data such as content objects. It can use any combination of database, LDAP server and SAP System to store user management data. As user-related data is sensitive data, you should protect all communication channels to user data stores.

There are also communication channels between the Portal Server and any backend systems used for providing content to display in the portal. Depending on the nature of the data passed from the backend systems to the Portal Server, these communication channels should also be protected. For example, the Portal Server can connect to SAP Systems using the remote function call (RFC) protocol. These connections can be secured using Secure Network Communications (SNC).

The following table gives you a quick overview of where to find detailed documentation on securing the communication channels shown in the diagram.

Connection	Secure Protocol	Documentation
Web browser ↔ Web server	Secure Sockets Layer (SSL)	See Web server documentation.
Web server ↔ SAP J2EE Engine	SSL	See Web server documentation and SAP J2EE Engine extension documentation. If you are using Microsoft Internet Information Server (IIS) as your Web server, see the document Enabling SSL redirection with the ISAPI module [Page 103].
Web browser ↔ SAP J2EE Engine	SSL	See the document Configuring the Use of SSL on the SAP J2EE Engine [Page 103].
Portal Server ↔ Database	SSL	No documentation currently available.
Portal Server ↔ LDAP Directory	SSL	SSL Between the User Management Service and an LDAP Directory [Page 49]

Portal Server ↔ SAP R/3 System	Secure Network Communications (SNC)	Configuring SNC Between User Management Engine and SAP System [Page 52]

Unification Server

If you are using unification in your portal installation, we recommend that you configure SSL to the Unification Server. For details, see the section on *Secure Sockets Layer Support* in the *SAP Unification Server Administration Guide*.

Retrieval and Classification (TREX)

If you are using Retrieval and Classification (TREX) in your portal installation, there are three connections that you can secure. These are:

- Secure communication between the TREX preprocessor and portal Web server
- Secure communication between the TREX Web server and TREX Java client (CM)
- Secure communication between the TREX Web server and TREX ISAPI Register (Windows only)

For more information, see Configuration of the TREX Security Settings [Page 63].

SSL Between the User Management Service and an LDAP Directory

Use

You can configure secure connections using the Secure Sockets Layer (SSL) protocol between the user management service and an LDAP directory. When SSL is used, the data transferred between the two parties (client and server) is encrypted.

The user management service uses server authentication for the SSL connection between the LDAP directories and the Portal Server. This means that the server (in this case, the LDAP directory) provides its identity to the client (in this case the user management service) using a certificate, but the client does not provide its identity to the server.

Once the secure connection is set up, the user management service binds to the LDAP directory with the LDAP protocol using user ID and password. This user ID and password, and all other data that is passed between the two parties is encrypted.

Prerequisites

To configure an SSL connection between the user management service and an LDAP directory, the following is required:

- You have generated a certificate for the LDAP directory server. This can either be a self-signed certificate or a certificate issued by a certification authority. The certificate should be in DER format. Read the documentation of your directory server vendor for instructions on how to generate a certificate.
- You have configured the directory server to support SSL. Again, read the directory server documentation for instructions.
- We highly recommend that you configure an SSL connection between the portal Web server and the browser.

Process

1. You configure the Java application server [Page 50].

   This involves creating a keystore on the Java application server and importing the certificate of the LDAP directory server into this keystore. This ensures that the Java application server trusts the LDAP directory server. You also need to make additional settings on the Java application server.

2. You configure the user management service [Page 51].

   Here you configure the user management service to use SSL and provide the secured port of the LDAP server.

Configuring SAP J2EE Engine for SSL to an LDAP Directory

Import root CA certificate of LDAP server certificate into trust store of SAP J2EE Engine

For the following step, you require the `keytool.exe` tool that comes with the Java Development Kit (JDK). You can find this tool at `usr\java\jre\bin`.

1. Use the `keytool.exe` tool to import the root CA certificate of the LDAP server certificate into the trust store of the SAP J2EE Engine by entering the following command at the command prompt:

   ```
   keytool –import –file <ldap_certificate> -keystore
   <path_to_trust_store> -storepass <password_for_trust_store> –
   alias ldapserver
   ```

   where `<ldap_certificate>` is the path and filename of the root CA certificate.

   The trust store used is the `cacerts` file located at `<java_home>/jre/lib/security/cacerts`. `<password_for_trust_store>` is the password for the `cacerts` file (`changeit` is the default password).

   The following command takes the root CA certificate located at `c:\rootcert.der` and imports it into the truststore `cacerts` with password `changeit` located in the current directory:

   ```
   keytool –import –file c:\rootcert.der -keystore cacerts
   -storepass changeit –alias ldapserver
   ```

Install libraries

You need to install some libraries that are not shipped with the Enterprise Portal as they are subject to German export control regulations. These are available on the SAP Service Marketplace.

2. Download the files `iaik_jce.jar` and `iaik_ssl.jar and` copy them to `<J2EE_Server>` [Page 103]`/additional-lib`.

   You can download these files from the SAP Service Marketplace at *http://service.sap.com/swcenter* → *SAP Cryptographic Software* → *SAP JAVA Cryptographic Toolkit* using your customer user ID. Use the program `sapcar.exe` to extract the jar files from the downloaded car file by entering the following at the command prompt:

   ```
   sapcar –xvf <car_file>
   ```

You also need to do the following:

3. Make sure that the files `tc_sec_api.jar` and `tc_sec_core.jar` are in the <J2EE_Server> [Page 103]`/additional-lib` folder.
4. Open the file <J2EE_Server> [Page 103]`/managers/library.txt` and make sure that the following lines exist:

   ```
   library sap_security tc_sec_api.jar;tc_sec_core.jar

   reference sap_security logging
   ```

5. Open the file <J2EE_Server> [Page 103]`/managers/reference.txt` and make sure that the following lines exist:

   If you are using SAP User Management Engine in the Enterprise Portal:

   ```
   reference irj library:IAIKSecurity
   reference irj library:sap_security
   reference irj library:logging
   ```

   If you are using SAP User Management Engine stand alone:

   ```
   reference UserManagementEngine library:IAIKSecurity
   reference UserManagementEngine library:sap_security
   reference UserManagementEngine library:logging
   ```

Debugging

To switch on SSL tracing, add the following system property to the VM:

`-Dcom.sap.security.ssl.debug=true`

Debugging information is then written to the console ouput.

Configuring User Management Service for SSL to an LDAP Directory

If you are using SAP User Management Engine in the Enterprise Portal

1. Start the User Management configuration tool by choosing *System Administration* → *System Configuration* → *UM Configuration.*
2. Choose the *LDAP Server* tab.
3. If the connection data for the LDAP server has not already been entered, enter it now. See *Enterprise Portal Administration Guide* → *Portal* → *User Management Configuration* → *Configuration of Data Sources Used for User Management*.
4. In the *Port* field, specify the SSL port of the LDAP server.

   The default SSL port for LDAP is 636.
5. Select the *Use SSL* checkbox.
6. Choose *Save*.
7. Restart the Java application server.

If you are using SAP User Management Engine as a stand alone component

1. Open the `sapum.properties` file and set the following properties as indicated:

   `ume.ldap.access.ssl= true`
   Change the following connection properties to the appropriate values to get an SSL connection to the LDAP server, for example, you need to change the port to the SSL port of the LDAP server.

   ```
   ume.ldap.access.server_name=<servername>
   ume.ldap.access.server_port=<SSL_Port>
   ume.ldap.access.user=<user>
   ume.ldap.access.password=<password>
   ```

2. Restart the Java application server.

Configuring SNC Between User Management Engine and SAP System

Use

This section describes how to configure Secure Network Communications (SNC) on connections between SAP User Management Engine (UME) and SAP Systems. You can use this procedure to activate SNC both on connections between UME and an SAP System that it uses as a persistence store and between UME and SAP Systems to which UME replicates user data.

We strongly recommend that you protect these types of connections with SNC as sensitive user data is passed over these connections.

The following procedure applies both for UME integrated with SAP Enterprise Portal and used as a standalone component with other solutions. It describes a scenario where the SAP Cryptographic Library is used as the security product.

You have a choice between the following two scenarios for configuring the use of SNC between UME and an SAP System:

- You can create a single Personal Security Environment (PSE) that is shared by UME and the SAP Systems by copying it to each of the server hosts. This option is better if you have only one UME and one SAP System, for example, in a test environment. It is the simpler option.
- You can create individual PSEs for each of the system components. This option is more complicated to configure, but is recommended if you intend to configure UME for SNC with several SAP Systems.

The configuration steps for both of these scenarios are described below. This documentation focuses on configuration required in UME. For detailed documentation on how to configure the SAP System, see:

- SNC *User's Guide*: This guide provides a full description of how to use SNC with SAP Systems. You can find this guide on the SAP Service Marketplace at: *http://service.sap.com/security* → Security in Detail → *Secure System Management.*
- SAP Web Application Server documentation at *SAP Web Application Server* → *Security (BC-SEC)* → *Secure Network Communications (BC-SEC-SNC).*

Prerequisites

- You must be able to receive the SAP Cryptographic Library as stated by the German export regulations. The library is available on the SAP Service Marketplace at http://service.sap.com/swcenter.

- You must know your SNC naming convention and the SNC names for the application server and UME.

  The server may use additional PSEs for other purposes, for example, UME also has a PSE that it uses to digitally sign SAP logon tickets. To avoid naming conflicts, use a unique Distinguished Name for UME's SNC PSE.

Procedure

Depending on the scenario you use, see either:

- Configuring SNC When Using a Single PSE [Page 53]
- Configuring SNC When Using Individual PSEs [Page 54]

Configuring SNC When Using a Single PSE

Purpose

In this case, you create a single PSE that is used by both SAP User Management Engine (UME) and the SAP System application server. If the SAP System already has an SNC PSE you can copy this PSE to UME. Otherwise you create a new PSE on UME and copy it to the SAP System.

Prerequisites

The SAP R/3 System is already SNC-enabled. For details, see the *SNC User's Guide*, which you can find on the SAP Service Marketplace at http://service.sap.com/security → *Security in Detail* → *Secure System Management.*

Process Flow

1. Make sure that the SAP Cryptographic Library is available on the host on which UME is installed. If it is not available, install it [Page 54].
2. Set the `SECUDIR` environment variable to the location in which you wish to store the SNC PSE file. Then restart the UME host.
3. If the SAP System already has an SNC PSE file, copy the file to the UME machine [Page 55]. Otherwise, create a PSE file for UME [Page 55] and copy it to the SAP System application server to the directory defined in the `SECUDIR` environment variable.
4. Create credentials for UME [Page 56].
5. Set SNC properties for SAP System in User Management Engine [Page 60].
6. Configure the SAP R/3 System [Page 61] to allow an SNC protected connection with UME.

Configuring SNC When Using Individual PSEs

Purpose

In this case, you generate a separate PSE for the application server and SAP User Management Engine (UME) and exchange their public keys so that the two components can communicate with each other using SNC.

Prerequisites

The SAP R/3 System is already SNC-enabled. For details, see the *SNC User's Guide*, which you can find on the SAP Service Marketplace at http://service.sap.com/security → *Security in Detail* → *Secure System Management.*

Process Flow

1. Make sure that the SAP Cryptographic Library is available on the host on which UME is installed. If it is not available, install it [Page 54].
2. Set the `SECUDIR` environment variable to the location in which you wish to store the SNC PSE file. Then restart the UME host.
3. Create a PSE file for UME [Page 55].
4. Create credentials for UME [Page 56].
5. Exchange the servers' public-key certificates [Page 58].
6. Set SNC properties for SAP System in User Management Engine [Page 60].
7. Configure the SAP R/3 System [Page 61] to allow an SNC protected connection with UME

Step-By-Step Procedures

In the following section, you can find step-by-step instructions for the different steps required to configure SNC between SAP User Management Engine and a SAP System.

Installing SAP Cryptographic Library

Prerequisites

You have access to the SAP Cryptographic Library.

Procedure

Copy the SAP Cryptographic Library (`sapcrypto.dll`) for your platform, the configuration tool (`sapgenpse.exe`), and the corresponding license ticket (`ticket`) to local directories on the machine on which User Management Engine is installed.

You can download the Cryptographic Library from the SAP Service Marketplace at *http://service.sap.com/swcenter* → *SAP Cryptographic Software* → *SAP Cryptographic Library <your platform>* using your customer user ID. See also SAP Note 397175.

Copying SAP System's PSE to UME (Single PSE)

Prerequisites

- The SAP System already has an SNC PSE.
- On the SAP User Management Engine (UME) host, the environment variable `SECUDIR` is set to the location where the PSE is stored.

Procedure

1. Export the SAP System's SNC PSE using transaction *STRUST*.
2. Copy the exported PSE to the `SECUDIR` directory on the UME machine.

Creating PSE for UME

Use

Use the command `get_pse` to generate a PSE for SAP User Management Engine (UME). This PSE includes the public and private key pair and a public-key certificate. If you are using a trusted CA, then you can also use the `get_pse` command to generate a certificate request. The following procedure describes how to create a PSE with a self-signed certificate.

Prerequisites

- The SAP Cryptographic Library is installed on the UME host.

Procedure

1. On the UME host, set the `SECUDIR` environment variable to the location where you want to store the PSE file by entering the following at the command line prompt:

   ```
   set SECUDIR=<path_to_PSE_file>
   ```

   We suggest that you store the PSE in the same directory in which you installed the SAP Cryptographic library:

   ```
   <DRIVE>:\snc\SAPCryptoLib
   ```

   Example:

   ```
   set SECUDIR=C:\snc\SAPCryptoLib
   ```

   As an alternative, you can change to the desired directory and set `SECUDIR` as indicated below:

   ```
   cd C:\snc\SAPCryptoLib
   set SECUDIR=.
   ```

   By using this technique, you can avoid problems such as case-sensitivity or shortened directory names in Windows that use the tilde character (~).

2. Use the command line tool `sapgenpse` to create a PSE by entering the following at the command line prompt:

   ```
   sapgenpse get_pse -p <pse_file> -noreq -x <PIN>
   <Distinguished_Name>
   ```

   where

Parameter	Description	Allowed Values
`<pse_file>`	Path and file name for UME's PSE.	Path description (in quotation marks, if spaces exist)
`<Distinguished_Name>`	Distinguished Name (DN) for the server. The Distinguished Name is used to build UME's SNC name.	Character string (in quotation marks, if spaces exist). The DN must be in the following format: `CN=<common_name>, OU=<organizational_unit>, O=<organization>, C=<country>.` The DN used for the SNC PSE must be different to the DN used for the certificate for signing logon tickets.
`<PIN>`	PIN that protects the PSE	Character string

```
sapgenpse get_pse -p c:\snc\SAPCryptoLib\UME.pse –noreq –x
abcpin "CN=UME, OU=MYCOMPANY, O=SAP-AG, C=DE"
```

This generates a PSE file located at `c:\snc\SAPCryptoLib\UME.pse` which includes a public key, private key and self-signed certificate. For details on the `sapgenpse.exe` tool, see the SAP Web Application Server documentation at *SAP Web Application Server → Security (BC-SEC) → Secure Network Communications (BC-SEC-SNC) → Configuring the Use of the SAP Cryptographic Library for SNC → Configuring SNC for Using the SAPCRPYTOLIB Using SAPGENPSE*

Result

UME's PSE is created in the directory you specified.

Check the contents of the directory at the operating system level to make sure the PSE was created in the correct location before proceeding with the next step.

Creating Credentials for UME

Use

To be able to access its PSE at run-time, SAP User Management Engine (UME) requires active credentials, which you create by using the configuration tool to "open" UME's PSE.

The credentials are located in the file `cred_v2` in the directory specified in the environment variable `SECUDIR`. Make sure that **only the user under which the J2EE Engine runs** has access to this file (including read access).

It is also very important to create the credentials for the **user who runs the J2EE Engine's processes**.

Prerequisites

- The server possesses a PSE and you know where it is located.
- You know the user that runs the J2EE Engine's processes.

  If you are unsure which user this is, see Checking the Java Servlet Engine's User [Page 58]. Make sure that **only this user** has access to the credentials file!
- The environment variable `SECUDIR` is set to the location where the PSE is stored for the user under which the servlet engine is running.

Procedure

1. Open a shell and go to the `SECUDIR` directory (make sure the `SECUDIR` environment variable is active).
2. Enter the following at the command line prompt to open the server's PSE and create credentials:

   ```
   sapgenpse seclogin –p <PSE_name> -x <PIN> -O <J2EE_Engine_user>
   ```

   where

Parameter	Description	Allowed Values
`<PSE_name>`	Path and file name for the server's PSE	Path description (in quotation marks, if spaces exist)
`<PIN>`	PIN that protects the PSE	Character string
`<J2EE_Engine_user>`	User for which the credentials are created. (The user that the servlet engine is running under). If you are unsure which user this is, see Checking the Java Servlet Engine's User [Page 58].	

   ```
   sapgenpse seclogin -p UME.pse –x abcpin -O SYSTEM
   ```

   In this example, the command opens the UME's PSE that is located in the `SECUDIR` directory in the file `UME.pse`, and creates credentials (`cred_v2`) for the user `SYSTEM` in the `SECUDIR` directory. The PIN that protects the PSE is `abcpin`.
3. Adjust the file permissions for the PSE (`<file_name>.pse`) and credentials file (`cred_v2`) so that the server's user can access them at run-time.

Result

The credentials file (`cred_v2`) for the user specified with the `-O` option is created in the `SECUDIR` directory. This user can then access the credentials at run-time.

Checking the Java Servlet Engine’s User

Use

Use this procedure to determine which user the Java servlet engine of the Portal Server uses to run its processes so that you can create credentials for the correct user.

Procedure

On the Portal Server machine:

1. Choose *Start* → *Control Panel* → *Administrative Tools* → *Services*.

   The *Services* screen appears.

2. Select the service for the Java servlet engine. For SAP J2EE Engine, the service is called *SAP J2EE Engine.*

3. Right-click on the service and choose *Properties* → *Log On*.

   The *Properties* screen for the service appears.

   - If *Local System Account* is selected, then the user used by the Java servlet engine to run its processes is SYSTEM.
   - Otherwise, the user used by the Java servlet engine to run its processes is the user displayed in the *This Account:* field.

Exchanging the Servers’ Public-Key Certificates

Use

Use the following procedure if each server possesses an individual PSE. In this case, you must exchange the servers' public-key certificates so that they can identify each other using SNC. You must import the partner's certificates on each host for each set of communication partners.

For details on the `export_own_cert` and `maintain_pk` options of the configuration tool, see SAP Web Application Server documentation at *SAP Web Application Server* → *Security (BC-SEC)* → *Secure Network Communications (BC-SEC-SNC).*

Prerequisites

- Both SAP User Management Engine (UME) and the SAP System application server possess their own PSE and you know where they are located.
- `SECUDIR` has been set to the location where the credentials are to be stored.

Procedure

On the UME host:

1. Export UME’s public-key certificate by executing the following configuration tool command line:

   ```
   sapgenpse export_own_cert -o <output_file> -p <PSE_name>
   [-x <PIN>]
   ```

The following command line exports UME's certificate and stores it in the file `UME.crt` in the directory in which you called the command:

```
sapgenpse export_own_cert -o UME.crt -p UME.pse -x abcpin
```

2. Make the certificate available to the SAP System application server. For example, copy it to a shared directory in the file system.

On one of the SAP System application server hosts:

3. Export the application server's public-key certificate by executing the following configuration tool command line:

```
sapgenpse export_own_cert -o <output_file> -p <PSE_name>
[-x <PIN>]
```

The following command line exports the application server's certificate and stores it in the file `D:\usr\sap\ABC\DVEBMGS28\sec\ABC.crt`:

```
sapgenpse export_own_cert -o D:\usr\sap\ABC\DVEBMGS28\sec\
ABC.crt -p D:\usr\sap\ABC\DVEBMGS28\sec\ABC.pse -x abcpin
```

4. Make the certificate available to UME. For example, copy it to a shared directory in the file system.

On each of the SAP System application server hosts:

5. Import UME's certificate into the application server's certificate list using the following configuration tool command line:

```
sapgenpse maintain_pk [<additional options>] [-a <cert_file>]
[-d <number>] -p <PSE_name> [-x <PIN>]
```

The following command line imports the previously exported UME's certificate (now located at `D:\usr\sap\ABC\DVEBMGS28\sec\UME.crt`) into the application server's certificate list:

```
sapgenpse maintain_pk -a
D:\usr\sap\ABC\DVEBMGS28\sec\UME.crt -p
D:\usr\sap\ABC\DVEBMGS28\sec\ABC.pse -x abcpin
```

On the UME host:

6. Import the application server's certificate into UME's certificate list using the following configuration tool command:

```
sapgenpse maintain_pk [<additional options>] [-a <cert_file>]
[-d <number>] -p <PSE_name> [-x <PIN>]
```

The following command line imports the previously exported application server's certificate (now located at `c:\SAP_J2EEEngine6.20\SAPCryptoLib\ABC.crt`) into UME's certificate list:

```
sapgenpse maintain_pk -a
"c:\SAP_J2EEEngine6.20\SAPCryptoLib\ABC.crt" -p
"c:\SAP_J2EEEngine6.20\SAPCryptoLib\UME.pse" -x abcpin
```

Result

The two servers have exchanged their public-key certificates so that they can identify each other when using SNC connections.

Setting UME Properties for SNC

Use

To activate the SNC connection between SAP User Management Engine (UME) and an SAP System, you must set properties relating to the SAP System in the UME properties file, `sapum.properties`.

For details on how to change user management properties, see *SAP Enterprise Portal Administration Guide → Portal → System Administration → User Management Configuration → User Management Properties.*

Procedure

1. In the UME properties file, add the following properties to the properties already maintained for the SAP System:

Property Name	Description
`ume.r3.connection.<adapterID>.user`	Service user in SAP System. For more details, see Requirements for Service User Used to Connect to SAP Systems [Page 61].
`ume.r3.connection.<adapterID>.snc_lib`	Location of cryptographic library.
`ume.r3.connection.<adapterID>.snc_myname`	SNC name of SAP User Management Engine. This is the distinguished name in the UME PSE in the following format: `p:<distinguished_name_of_UME_PSE>`
`ume.r3.connection.<adapterID>.snc_partnername`	SNC name of SAP System. This is the distinguished name in the SAP system's SNC PSE in the following format: `p:<distinguished_name_of_R/3_PSE>`
`ume.r3.connection.<adapterID>.snc_mode`	To activate SNC, this must be set to `1`.

The default value for `<adapterID>` is `master`, however you can change it by assigning a different value to the property `ume.logon.r3master.adapterid`. For example,

`ume.logon.r3master.adapterid=ABC`

The following is an example of an excerpt of `sapum.properties`:

```
ume.r3.connection.master.client=123
ume.r3.connection.master.r3name=ABC


ume.r3.connection.master.user=sapjsf2
ume.r3.connection.master.snc_lib=c:\snc\SAPCryptoLib\sapcrypto.dll
ume.r3.connection.master.snc_myname=p:CN=UME, OU=MYOU, O=MYCOMPANY,
C=DE
```

```
ume.r3.connection.master.snc_partnername=p:CN=ABC, OU=MYOU,
O=MYCOMPANY, C=DE
ume.r3.connection.master.snc_mode=1
```

Requirements for Service User Used to Connect to SAP Systems

To connect from SAP User Management Engine (UME) to an SAP System using RFC (with or without Secure Network Communications), you need to specify a service user with which to establish the connection. You must create this service user in the SAP System and it must fulfil the requirements listed below.

- **User ID:** We recommend that you use the user ID SAPJSF for the service user.
- **User Type:** The user must be of type *communication user* (or *CPIC* in older releases).
- **Authorization:** The user requires authorizations for read access to user data, for authenticating remote users, and RFC authorizations.

  As of Release 6.20, SAP Web Application Server is shipped with two roles that provide the required authorizations:

  - *SAP_BC_JSF_COMMUNICATION_RO* provides all authorizations for read access to user data, for authenticating remote users, and several low-level RFC authorizations.
  - *SAP_BC_JSF_COMMUNICATION* is the same as the above role, but additionally provides authorization to modify and delete all user-related data.

  We recommend using *SAP_BC_JSF_COMMUNICATION_RO*.
- **SNC name:** If the connection is secured with SNC, the user must be assigned to the SNC name used by the SAP System. To do this, in transaction SU01, on the SNC tab, enter the SNC name of the SAP System. You can find the SNC name of the SAP System in table USRACL.

See Also:

For SAP Systems with release higher than or equal to 6.20, see *Integration of the Security Functions of the ABAP Stack and J2EE Stack*. You can find this document on SAP Service Marketplace at http://service.sap.com/security → *Security in Detail -> Hot Topic: J2EE*.

Configuring SAP R/3 System for SNC

Use

Use this procedure to configure the SAP R/3 System to allow an SNC protected connection with SAP User Management Engine (UME).

Prerequisites

The SAP R/3 System is already SNC-enabled. For details, see the *SNC User's Guide*, which you can find on the SAP Service Marketplace at http://service.sap.com/security → *Security in Detail* → *Secure System Management.*

Procedure

There are two types of access control lists (ACLs) that you need to maintain in the SAP R/3 System: a system ACL and a user ACL.

Add UME SNC Name to System Access Control List

After you add the UME SNC name to the system ACL, the SAP R/3 System allows UME to establish an SNC protected connection to the SAP System application server.

2. In the SAP R/3 System, start table maintenance for table VSNCSYSACL (for example, use transaction SM30, enter the table name, and choose *Maintain*).
3. For *Type of ACL entry* enter *E*.
4. Choose *New entries*.
5. Enter data in the fields as follows:

Field Name	Entry	Comments
System ID	Leave this field blank	
SNC Name	p:<distinguished_name_of_ UME_PSE>	The distinguished name is the one you specified when you created the PSE for UME [Page 55]. This entry should have the same value as `ume.r3.connection.master.snc_myname` in the UME properties file [Page 60].

6. Activate the *Entry for RFC activated* indicator.
7. Save your entries.

Add UME SNC Name to User Access Control List

This allows portal users to connect to the SAP System using UME's SNC connection. The users themselves are explicitly authenticated at connection time.

1. In the SAP R/3 System, start table maintenance for table USRACLEXT (for example, use transaction SM30, enter the table name, and choose *Maintain*).
2. Choose New entries.
3. Enter data in the fields as follows:

Field name	Entry	Comments
User	asterisk symbol (*)	The wildcard entry allows all users to be able to connect to the SAP system using the SNC protected connection from UME.
Seq. number		Not required
SNC name	p:<distinguished_name_of_ UME_PSE>	The distinguished name is the one you specified when you created the PSE for UME [Page 55]. This entry should have the same value as `ume.r3.connection.master.snc_myname` in the UME properties file [Page 60].

4. Save your entries.

Troubleshooting

If you experience problems with the SNC connection, do the following:

- Check whether the SECUDIR environment variable is set correctly.
- If you have SECUDE PSE management on the computer, log off (otherwise the two credentials may interfere with each other).
- Check that you created credentials for the correct user.
- Set the TRACE option for the JCo connection by entering the following in the UME properties file, `sapum.properties`:

  ```
  ume.r3.connection.<adapterID>.trace=1
  ```

  The RFC layer will then create log files called `dev_rfc.trc` and `rfc_XXXXX_XXXXX.trc` (where X denotes a digit).

Configuration of the TREX Security Settings

Purpose

Retrieval and Classification (TREX) enables you to configure secure communication between the Enterprise Portal (portal Web server, TREX Java client (CM)) and the TREX components (TREX Preprocessor, TREX Web server, TREX ISAPI Register).

Secure Communication Between TREX Components and the Portal

Purpose

The Secure Sockets Layer protocol (SSL protocol) with client authentication is used for secure communication between the TREX components and the portal. SSL with client authentication ensures:

- Confidentiality – The data is transmitted in encoded form and cannot be intercepted.
- Data integrity – The recipient can be sure that the transmitted data cannot be changed during the transmission.
- Authentication – The communication partners know with whom they are communicating.

Secure communication is based on the use of electronic certificates. A certificate contains the public key of the owner and information on the owner, for example, his or her name (common name), organizational unit, or e-mail address. Certificates are issued by a certification authority (CA) that confirms the identity of the certificate's owner. The public and private certificates of the certificate owner are kept in a keystore (Personal Security Environment or PSE) that is protected by a password.

The two communication partners can then encrypt their messages before sending them. Administrators provide the necessary certificates. They also configure the security settings for the components and modify security-relevant parameters in the TREX configuration files.

Prerequisites

- In your enterprise, you have built up a public key infrastructure with your own CA that issues certificates.

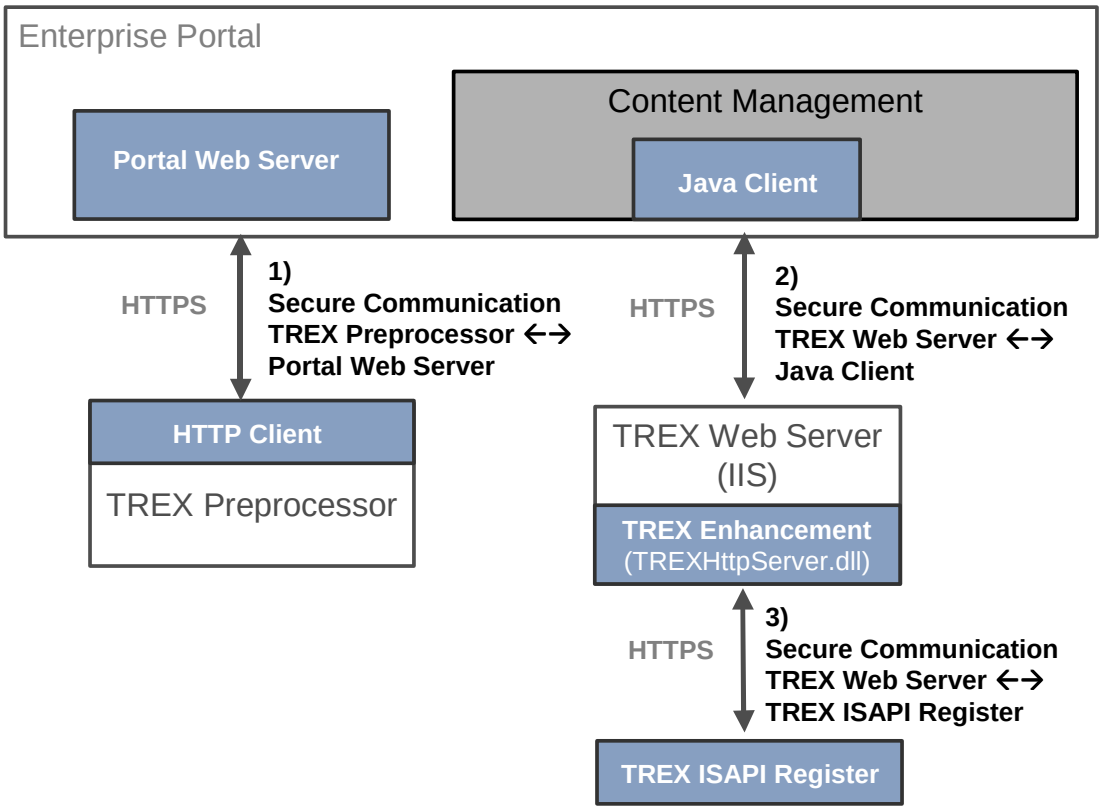
  or

- You are working with any organization that offers the issuing of certificates.

Secure Communication Areas

There are three areas of secure communication between TREX components and the portal.

1. Secure communication between the TREX preprocessor and portal Web server [Page 75]
2. Secure communication between the TREX Web server and TREX Java client (CM) [Page 80]
3. Secure communication between the TREX Web server and TREX ISAPI Register (Windows Only) [Page 96]



You only have to configure secure communication between the TREX Web server and the TREX ISAPI register for a TREX installation on Windows. On UNIX, TREX uses the Apache Web server. This registers itself with the TREX name server when it starts, and does not have an ISAPI extension.

Usage of SAP Cryptography Tools

Purpose

You use the following tools for configuring and managing secure communication:

SAPGENPSE

The *SAPGENPSE* cryptography tool consists of the following files:

Windows	UNIX
`sapcrypto.dll` (library)	`libsapcrypto.<ext>`, for example, for the operating system SUN OS 5.8,
`sapgenpse.exe` (executable file)	`sapgenpse` (executable file).
`ticket` (license ticket)	`ticket` (license ticket)

You use *SAPGENPSE* to configure secure communication between the TREX preprocessor and the portal Web server and between the TREX Web server and the TREX ISAPI register. You download the *SAPGENPSE* cryptography tool as part of the *SAP Cryptographic Library* in the SAP Service Marketplace.

SAP Crypto Manager

The *SAP Crypto Manager* consists of the following files:

- `iaik_jce.jar`
- `iaik.jsse.jar`
- `iaik_ssl.jar`
- `w3c_http.jar`

You can use the *SAP Crypto Manager* to configure secure communication between the TREX Web server and the TREX Java client in Content Management. You obtain the *SAP Crypto Manager* as part of the *SAP Java Cryptographic Toolkit* in the SAP Service Marketplace.

You also need the files listed below. They can be found in the `C:\SAP_J2EEngine6.20\alone\additional-lib` directory in a complete portal installation.

- `tc_sec_api.jar`
- `tc_sec_core.jar`
- `tc_sec_jni.jar`

Internet Information Services (Microsoft© Management Console for IIS)

You find this tool under Start → Settings → *Control Panel* → *Administrative Tools* → *Internet Services Manager*.

To manage the security settings, choose *Internet Information Services* → *<your server>* → *SAP_TREX* → *TREXHttpServer* → *Properties* (secondary mouse-button click) → *Directory settings.*

Microsoft© Internet Explorer

To manage the security settings, choose *Tools* → *Internet Options* → *Content* → *Certificates* in your Internet Explorer.

See also:

Downloading the SAP Cryptographic Library

Use

You download the *SAPGENPSE* cryptography tool as part of the *SAP Cryptographic Library* in the SAP Service Marketplace.

Prerequisites

- You have authorized access to the SAP Service Marketplace with a SAP s-user ID.
- You have installed the *SAP Download Manager* in your system.

  For more information on downloading, installing, and configuring the SAP download manager, see under *Download Tools* in the *Software Distribution Center* (`http://service.sap.com/swcenter`) in the SAP Service Marketplace.
- You have installed the SAP archiving tool *SAPCAR*.

  For more information on downloading, installing, and configuring *SAPCAR*, see SAP Note 212876: *The new archiving tool SAPCAR*.

Procedure

1. Start your Web browser and navigate to the page `http://service.sap.com/swcenter`.
2. Log on with your SAP s-user ID and navigate to *Download* → *SAP Cryptographic Software*.
3. Confirm your acceptance of the export regulations.

   For more information on the export regulations, see SAP Note 397175: *SAP Cryptographic Software – Export check*.
4. In the *SAP Download Area*, select the operating system that you require.

   `Windows 2000:` `SAP Cryptographic Library Microsoft Win32 for x86/IA32 for MS Win32`

   `Sun Solaris:` `SAP Cryptographic Library Sun Solaris for SPARC for Sun Solaris`
5. Select the files that you require and download them using the SAP Download Manager.

6. Store the files in a temporary directory in your system, and unpack them using the SAP archiving tool *SAPCAR*.

Contents of the SAP Cryptographic Library

The *SAP Cryptographic Library* installation package includes the following files:

Windows	UNIX
`sapcrypto.dll` (Library) **`sapgenpse.exe`** (executable file)	`libsapcrypto.<ext>` (Library) `libsapcrypto.<ext> (Library)`, for example, `libsapcrypto.so` for the operating system OS 5.8, **`sapgenpse`** (executable file)
`ticket` (licence ticket)	`ticket` (licence ticket)

Result

When you have copied the files of the *SAP Cryptographic Library* installation package to your computer, you can install, start, and configure *SAPGENPSE*.

Configuring SAPGENPSE for Use

Use

If you are configuring and using the cryptography tool *SAPGENPSE*, you should be aware of which files you require and know where they are stored. These files are not only required initially for the authentication of the communication partner. They are also required during data transmission in order to encode the data and ensure data integrity. This means that the files must be stored in particular places so that the system can find them and access them at runtime.

Prerequisites

You have downloaded the SAP Cryptographic Library from the SAP Service Marketplace and unpacked the contained data.

Required Files

Windows	UNIX
`sapcrypto.dll` (Library) **`sapgenpse.exe`** (executable file)	**`libsapcrypto.<ext>`,** `for example, libsapcrypto.so` for the operating system SUN OS 5.8, `sapgenpse` (executable file).
`ticket` (licence ticket)	**`ticket`** (licence ticket)
`SAPSSLS.pse` Keystore for server certificates	**`SAPSSLS.pse`**
`SAPSSLC.pse` Keystore for client certificates	**`SAPSSLC.pse`**
`SAPSSLA.pse` Anonymous keystore	**`SAPSSLA.pse`**

You create the keystores `SAPSSLS.pse`, `SAPSSLC.pse`, and `SAPSSLA.pse` using the cryptography tool *SAPGENPSE*. These are **not** part of the *SAP Cryptographic Library* installation package.

Procedure

You configure the cryptography tool *SAPGENPSE* for use by achieving the following prerequisites:

- **Creating directories and environment variables**
  You create different directories and the environment variable SECUDIR on **Windows** and **UNIX** for storing the downloaded files and the keystores to be created.
- **Storing files in the recommended storage locations**
  You then store the files in question in the existing or newly created directories.

Windows: Recommended Storage Locations

Files	Storage Location
`sapcrypto.dll` **`sapgenpse.exe`**	System environment variable: **`SAPRETRIEVALPATH`** Directory: `C:\Program Files\SAP\TREX_6`  The system environment variable SAPRETRIEVALPATH and the directory `C:\Program Files\SAP\TREX_6` are created during the installation of TREX.
`ticket`	System environment variable: `SECUDIR` Directory: `C:\sec`  If the system environment variable SECUDIR and the directory `C:\sec` do not yet exist, you have to create them both.

`SAPSSLS.pse`
`SAPSSLC.pse`
`SAPSSLA.pse`
	System environment variable: `SECUDIR`
	Directory: `C:\sec`

Windows: Creating the Directory and Environment Variable

You have to reset the environment variable `SECUDIR` for the configuration of *SAPGENPSE*.

1. Create the directory **`c:/sec`**.
2. Choose *Start* → *Settings* → *Control Panel* → *System.*
3. Choose *Environment Variables* from the *Advanced* tab.
4. Choose *System Variables* and *New* in the *Environment Variables* screen.
5. Enter **SECUDIR** as the *variable name* and **`C:/sec`** as the variable value. Confirm with *OK*.
6. Restart your computer so that the new system variable `SECUDIR` is recognized by your operating system.

UNIX: Recommended Storage Locations for Files

Files	Storage Location
`sapgenpse` `libsapcrypto.<ext>, for example, libsapcrypto.so` for the operating system SUN OS 5.8	Environment variable: `LD_LIBRARY_PATH` Directory in which the shared libraries are stored on UNIX  If you started TREX using the `TREX` script, the environment variable `LD_LIBRARY_PATH` and the corresponding directory were created automatically.
`ticket`	Environment variable: `SECUDIR` Directory: `SAP_RETRIEVAL_PATH/../sec`  As a rule, the `SECUDIR` environment variable does not yet exist. You have to create the variable using the shell script for setting environment variables on UNIX.

`SAPSSLS.pse`
`SAPSSLC.pse`
`SAPSSLA.pse`
	Environment variable: `SECUDIR`
	Directory: `SAP_RETRIEVAL_PATH/../sec`

You create the keystores `SAPSSLS.pse`, `SAPSSLC.pse`, and `SAPSSLA.pse` using the cryptography tool *SAPGENPSE*. These are **not** part of the *SAP Cryptographic Library* installation package.

UNIX: Creating the Directory and Environment Variable

You have to set certain environment variables so that TREX runs. If you started TREX using the `TREX` script, the following environment variables were set automatically:

Environment variable	Description
`LD_LIBRARY_PATH`	Directory in which the shared libraries are stored.
`SAP_RETRIEVAL_PATH`	TREX installation directory
`PYTHONPATH`	Python installation directory

You have to reset the environment variable `SECUDIR` for the configuration of *SAPGENPSE*. There are two shell scripts in the TREX installation directory. You can use them to set an environment variable manually.

- `TREXSettings.sh` (Bourne-Shell `sh`, Bourne-again-Shell `bash`, Korn-Shell `ksh`)
- `TREXSettings.csh` (C-Shell `csh`)

Procedure

1. Log on with the user **`trexadm.`**
2. Create a directory called **`/sec`** parallel to the directory `SAP_RETRIEVAL_PATH.`
3. Go to the TREX installation directory.
4. Use a text editor to open the script for setting environment variables.

`TREXSettings.sh` (Bourne-Shell `sh`, Bourne-again-Shell `bash`, Korn-Shell `ksh`)

– `TREXSettings.csh` (C-Shell `csh`)

5. Define the directory for the environment variables by entering the following: **`SECUDIR=$SAP_RETRIEVAL_PATH/../sec`**
6. Enhance the specification for the existing entry `LD_LIBRARY_PATH=$SAP_RETRIEVAL_PATH` by adding **`:$SECUDIR`**
7. Enter **`export SECUDIR`** at the end so that the environment variable `SECUDIR` is valid for all opening windows.

`TREXSettings.sh` (Bourne-Shell `sh`, Bourne-again-Shell `bash`, Korn-Shell `ksh`)

```
#!/bin/sh
# TREX settings used by init script /etc/init.d/TREX
# configuration
SAP_RETRIEVAL_PATH=/usr/export/home/trexadm
SECUDIR=$SAP_RETRIEVAL_PATH/../sec

# create settings from configuration
PATH=$SAP_RETRIEVAL_PATH:$SAP_RETRIEVAL_PATH/Python/bin:$SA
P_RETRIEVAL_PATH/Apache/bin:$PATH:$SECUDIR
if [ -n "$LD_LIBRARY_PATH" ];
then
LD_LIBRARY_PATH=$SAP_RETRIEVAL_PATH:$SAP_RETRIEVAL_PATH/fil
ter:$LD_LIBRARY_PATH:$SECUDIR
else
LD_LIBRARY_PATH=$SAP_RETRIEVAL_PATH:$SAP_RETRIEVAL_PATH/fil
ter:$SECUDIR
fi
PYTHONPATH=$SAP_RETRIEVAL_PATH:$SAP_RETRIEVAL_PATH/Python/l
ib
PYTHONHOME=$SAP_RETRIEVAL_PATH/Python
export SAP_RETRIEVAL_PATH PATH LD_LIBRARY_PATH PYTHONPATH
PYTHONHOME
export SECUDIR
```

`TREXSettings.csh` (C-Shell `csh`)

```
#!/bin/csh
# TREX settings for cshell
# configuration
setenv SAP_RETRIEVAL_PATH /sapmnt/us0061/d/trexadm/TREX_SI2
# create settings from configuration
setenv PATH
${SAP_RETRIEVAL_PATH}:${SAP_RETRIEVAL_PATH}/Python/bin:${SA
P_RETRIEVAL_PATH}/Apache/bin:${PATH}
if ( ${?LD_LIBRARY_PATH} )
then
setenv LD_LIBRARY_PATH
${SAP_RETRIEVAL_PATH}:${SAP_RETRIEVAL_PATH}/filter:${LD_LIB
RARY_PATH}
else
setenv LD_LIBRARY_PATH
${SAP_RETRIEVAL_PATH}:${SAP_RETRIEVAL_PATH}/filter
endif
```

```
setenv PYTHONPATH
${SAP_RETRIEVAL_PATH}:${SAP_RETRIEVAL_PATH}/Python/lib
setenv PYTHONHOME ${SAP_RETRIEVAL_PATH}/Python
setenv SECUDIR ${SAP_RETRIEVAL_PATH}/../sec
```

8. Save the test script and close the text editor.
9. Now execute the relevant script.
   - Bourne-Shell `sh`, Bourne-again-Shell `bash`, Korn-Shell `ksh`:

     ```
     . TREXSettings.sh
     ```

   - C-Shell `csh`:

     ```
     source TREXSettings.csh
     ```

     Refer to the notes for the usage of keystores [Page 71].

Result

You use the *SAPGENPSE* cryptography tool to configure secure communication between the TREX preprocessor and the portal Web server and between the TREX Web server and the TREX ISAPI Register.

You start the cryptography tool *SAPGENPSE* using a prompt.

See also:

TREX Preprocessor and Portal Web Server [Page 75]

TREX Web Server and TREX ISAPI Register (Windows Only) [Page 96]

Usage of Keystores

Note the following when using keystores `SAPSSLA.pse`, `SAPSSLC.pse` and `SAPSSLS.pse`:

- **Keystore `SAPSSLS.pse` in `SECUDIR`**

  A `SAPSSLS.pse` keystore has to exist in the directory that you have defined under the SECUDIR environment variable. This is because the SAPCYPTOLIB (`sapcrypto.dll` library (Windows) or `libsapcrypto.<ext> (UNIX)`) is only initialized if a keystore in the form `SAPSSLS.pse` exists. Before creating and configuring new keystores, check the existing keystores and then take the appropriate action (generate the keystore `SAPSSLS.pse` or copy an existing keystore and rename it as `SAPSSLS.pse`).

- **Access sequence**

  `SAPCRYPTOLIB accesses the existing keystores in the following sequence:`

  1. `SAPSSL`**`A`**`.pse --> 2. SAPSSL`**`C`**`.pse --> 3. SAPSSL`**`S`**`.pse`

  If this keystore is available, you have to import your certificates to the keystore in the following order:

  1. Import the certificate to the keystore `SAPSSL`**`A`**`.pse`.
  2. Import the certificate to the keystore `SAPSSL`**`C`**`.pse`.

3. Import the certificate to the keystore SAPSSL**S**.pse.

This order is only valid for anonymous client authentication such as is configured between the portal Web server and the TREX preprocessor.

- **Format**

  For the keystore, write the part of the name that appears before the period in capitals (for example, **SAPSSL**.pse) and use lower case for the file extension (for example, SAPSSL.**pse**).

- **Initializing a keystore**
  After you have created a keystore, you have to initialize it for use. Enter the following command to do this:

  sapgenpse seclogin -p C:\sec\SAPSSLS.pse

  Note that the path specification in this example is only valid for Windows. On UNIX; specify the path according to the specification in the environment variable SECUDIR.

Command	Function
seclogin	Function of *SAPGENPSE* that you use to initialize a new keystore for use.
-p <path for environment variable **SECUDIR>\SAPSSLS.pse**	Specify the path and file name of the keystore that you want to initialize.  **c:\sec\SAPSSLS.pse** (Windows)

You are now asked to authorize this process by entering a password.

Prompt	Function/Entry
Please enter PIN:	Do **not** enter a value. Confirm with Return.
Please reenter PIN:	Do **not** enter a value. Confirm with Return.

- **Access permission for the keystores (Windows)**

  On Windows, you have to give the local system NT user access permission to the keystore files. Otherwise, the operating system cannot access the files. Enter the following command to do this:

  sapgenpse seclogin -p SAPSSLS.pse -O SYSTEM

Command	Function
-O SYSTEM	**SYSTEM** gives the local system NT user access to the keystore.

Result

You use the *SAPGENPSE* cryptography tool to configure secure communication between the TREX preprocessor and the portal Web server and between the TREX Web server and the TREX ISAPI Register.

You start the cryptography tool *SAPGENPSE* using a prompt.

See also:

TREX Preprocessor and Portal Web Server [Page 75]

TREX Web Server and TREX ISAPI Register (Windows Only) [Page 96]

Downloading the SAP Java Cryptographic Toolkit

Use

You download the *SAP Crypto Manager* as part of the *SAP JAVA Toolkit* from the SAP Service Marketplace.

Prerequisites

- You have authorized access to the SAP Service Marketplace with a SAP s-user ID.
- You have installed the *SAP Download Manager* in your system.

  For more information on downloading, installing, and configuring the SAP download manager, see under *Download Tools* in the *Software Distribution Center* (`http://service.sap.com`) in the SAP Service Marketplace.
- You have installed the SAP archiving tool *SAPCAR*.

  For more information on downloading, installing, and configuring *SAPCAR*, see SAP Note 212876: *The new archiving tool SAPCAR*.

Procedure

1. Start your Web browser and navigate to the page `http://service.sap.com/swcenter`.
2. Log on with your SAP s-user ID and navigate to *Download* → *SAP Cryptographic Software*.
3. Confirm your acceptance of the export regulations.

   For more information on the export regulations, see SAP Note 397175: *SAP Cryptographic Software – Export Check*.
4. In the *SAP Download Area*, choose *SAP JAVA Cryptographic Toolkit*.
5. Download the file using the *SAP Download Manager*.
6. Store the files in a temporary directory in your system, and unpack them using the SAP archiving tool *SAPCAR*.

The *SAP JAVA Cryptographic Toolkit* Installation Package Files

The *SAP JAVA Cryptographic Toolkit* installation package includes the following files:

- `iaik_jce.jar`
- `iaik.jsse.jar`
- `iaik_ssl.jar`

- `w3c_http.jar`

Result

You have copied the files from the *SAP JAVA Cryptographic Toolkit* to your host and can now install and start the *SAP Crypto Manager*.

Installing the SAP Crypto Manager

Use

You can use the *SAP Crypto Manager* to configure secure communication between the TREX Web server and the TREX Java client in Content Management for the portal.

Prerequisites

- You have downloaded the *SAP JAVA Cryptographic Toolkit* from the SAP Service Marketplace and unpacked the contained files.
- You have downloaded the following files from the directory `C:\SAP_J2EEngine6.20\alone\additional-lib` on the host on which the portal is installed:
  - `tc_sec_api.jar`
  - `tc_sec_core.jar`
  - `tc_sec_jni.jar`.

    You find these files in the directory `C:\SAP_J2EEngine6.20\alone\additional-lib` up to version 6.20 of the J2EE engine.

Procedure

To install the *SAP Crypto Manager*, proceed as follows:

1. Create a directory with the path **`C:\JavaProjects\SSL\CryptoManager`**.
2. Use a text editor to create a batch file called **`CryptoManager.bat`** with the following content:

```
set TCSEC_HOME=<Directory for "tc_sec"-JAR-files>
set IAIK_HOME=<Directory for "iaik"-JAR-files>
set JAVA_HOME=<Directory for the JDK>
"%JAVA_HOME%\bin\java.exe" (Java-Classpath) %DEBUG_PARAM% -cp
%TCSEC_HOME%\tc_sec_api.jar;%TCSEC_HOME%\tc_sec_core.jar;
%IAIK_HOME%\iaik_jce.jar com.sap.security.ssl.CryptoManager
```

   ```
   set TCSEC_HOME=C:\JavaProjects\SSL\CryptoManager
   set IAIK_HOME=C:\JavaProjects\SSL\CryptoManager
   set JAVA_HOME=C:\j2sdk1.4.1_01
   "%JAVA_HOME%\bin\java.exe" %DEBUG_PARAM% -cp
   ```

```
%TCSEC_HOME%\tc_sec_api.jar;%TCSEC_HOME%\tc_sec_core.jar;
%IAIK_HOME%\iaik_jce.jar com.sap.security.ssl.CryptoManager
```

In this example, the directory for the **tc_sec** JAR files and the directory for the **iaik** JAR files are identical.

3. Copy the following files to the directory `C:\JavaProjects\SSL\CryptoManager`.
   - `iaik_jce.jar, iaik.jsse.jar, iaik_ssl.jar` and `w3c_http.jar` from the *SAP JAVA Cryptographic Toolkit*
   - `tc_sec_api.jar, tc_sec_core.jar` and `tc_sec_jni.jar` from the portal directory for the J2EE Engine 6.20.

   If you created separate directories for the **tc_sec** and **iaik** JAR files, copy them to these directories appropriately.
4. Store the file `CryptoManager.bat` in the directory `C:\JavaProjects\SSL\CryptoManager` that you just created.

Starting the SAP Crypto Manager

Start the *SAP Crypto Manager* by double clicking on the batch file `CryptoManager.bat`.

Result

You can use the *SAP Crypto Manager* to configure secure communication between the TREX Web server and the TREX Java client in Content Management.

See also:

TREX Web Server and TREX Java Client (CM) [Page 80]

TREX Preprocessor and Portal Web Server

Purpose

The TREX preprocessor is responsible for preparing documents to be indexed by the TREX engines. The portal Web server transfers documents to be indexed to the preprocessor in the form of a URI that references the storage location of the document. The preprocessor resolves the URIs and uses HTTP to fetch the actual document from the repository in question. If this communication takes place using HTTPS, you have to configure anonymous client authentication between the preprocessor and the portal Web server.

Prerequisites

Use the *SAPGENPSE* cryptography tool to configure the preprocessor for HTTPS. This is a part of the *SAP Cryptographic Library*, which is the standard security product delivered by SAP for encryption functions in SAP systems. The *SAP Cryptographic Library* is available for download by authorized customers in the SAP Service Marketplace.

- We recommend that you install SAP Enterprise Portal and TREX on different hosts. The following is based on this standard scenario.
- You have access permission for the portal on which the Web server is running.

- You have installed and configured the *SAPGENPSE* cryptography tool on the host on which the TREX preprocessor is running. You can find more information on downloading and configuring *SAPGENPSE* under Usage of SAP Cryptography Tools [Page 64].

Process Flow

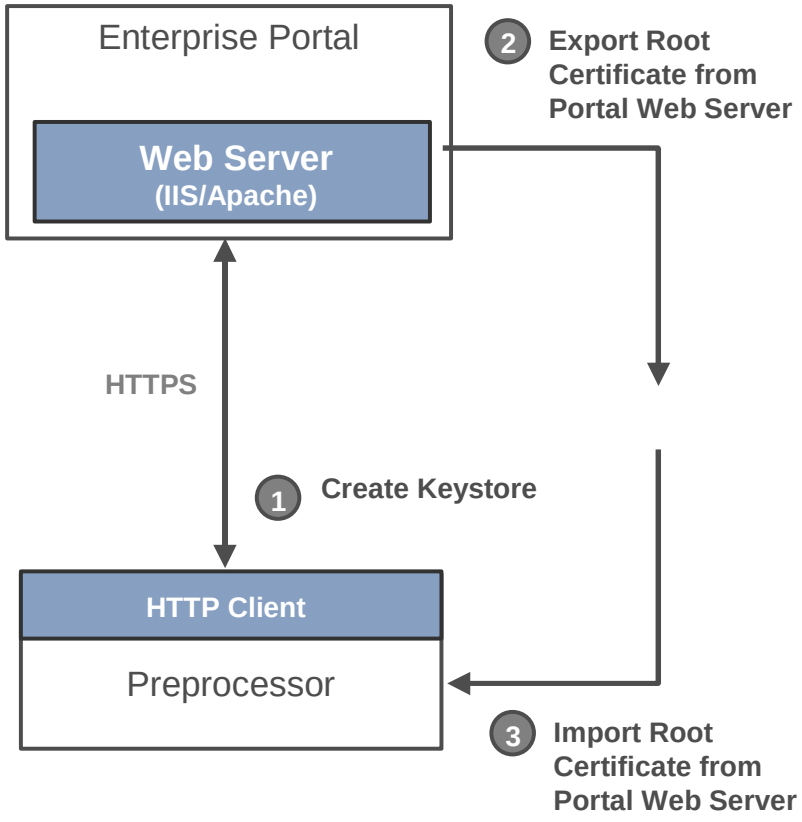
To configure anonymous client authentication between the preprocessor and the portal Web server, you need to

- Create a keystore in which you can create the root certificate of the portal Web server.
- Register the root certificate of the portal Web server with the TREX preprocessor

You carry out the following steps to do this:

1. Generate a keystore using *SAPGENPSE*
2. Export the root certificate from the portal Web server
3. Import the root certificate of the portal Web server

The figure below gives an overview of administrative tasks.



Result

You have configured anonymous client authentication between the TREX preprocessor and the portal Web server.

Generating a Keystore using SAPGENPSE

Use

You use the *SAPGENPSE* cryptography tool to generate a keystore in which you can store a certificate. You only need this keystore for storing the root certificate of the portal Web server. It is therefore not necessary that you send the generated certificate request to your CA.

Procedure

1. Open a prompt on the server on which the TREX preprocessor is installed.
2. Go to the directory in which the executable file **`sapgenpse.exe`** (Windows) or `sapgenpse` (UNIX) is located.
   - **Windows:** `C:\Program Files\SAP\TREX_6`
   - **`UNIX`:** `LD_LIBRARY_PATH`
3. Generate a new keystore by entering the following command:

   `sapgenpse gen_pse -p SAPSSLS.pse`

Command	Function
sapgenpse	Starts the *SAPGENPSE* cryptography tool.
gen_**pse**	Function of *SAPGENPSE* that you can use to generate a new keystore and a certificate request.
- p SAPSSLS.pse	You specify the file name of the keystore that contains the certificate here.

You are now asked to give more precise specifications on the certificates that you want to generate. Proceed according to the following table:

Prompt	Function/Entry
`Please enter PIN:`	Do **not** enter a value. Confirm with Return.
`Please reenter PIN:`	Do **not** enter a value. Confirm with Return.
`get_pse: Distinguished name of PSE owner:`	Specifies the distinguished name (DN) of the certificate owner. Make the following specifications: **`CN=myhost.mydomain, C=mycountry, S=mystate, O=mycompany, OU=mydepartment`**  **`CN=p64883.wdf.sap.corp, C=DE, S=BW, O=SAP-AG, OU=TREX`**

4. After you have created a keystore, you have to initialize it for use. Enter the following command to do this:

   `sapgenpse seclogin -p C:\sec\SAPSSLS. pse`

Command	Function
seclogin	Function of *SAPGENPSE* that you use to initialize a new keystore for use.

-p C:\sec\SAPSSLS.pse	Specify the path and file name of the keystore that you want to initialize.

You are now asked to authorize this process by entering a password.

Prompt	Function/Entry
`Please enter PIN:`	Do **not** enter a value. Confirm with Return.
`Please reenter PIN:`	Do **not** enter a value. Confirm with Return.

5. On Windows, you have to give the local system NT user access permission to the keystore files. Otherwise, the operating system cannot access the files. Enter the following command to do this:

   `sapgenpse seclogin -p SAPSSLS.pse -O SYSTEM`

Command	Function
-O SYSTEM	`SYSTEM` gives the local system NT user access to the keystore.

Result

You have created a keystore `SAPSSLS.pse` into which you can import the root certificate of the portal Web server and store it there.

Exporting the Root Certificate from the Portal Web Server

Use

You export the root certificate from the portal Web server using *Microsoft Internet Explorer*.

Procedure

1. Start SAP Enterprise Portal in your Internet Explorer on the host on which the portal is installed.
2. Choose *Tools* → *Internet Options* → *Content* → *Certificates* and then choose the tab page *Trusted Root Certification Authorities*.
3. Choose the root certificate that the portal trusts.
4. Choose *Export*. The *Certification Export Wizard* starts.
5. Choose *Next*. The *Export File Format* screen appears.
6. Choose the setting `BASE-64 encoded X.509 (.CER)` for the *export file format*.
7. Choose *Next*. The *File to Export* screen appears.
8. Under `filename`, name the file for the certificate to be exported in the format `<file_name>.cer`.
9. Choose *Browse.*
10. On the host on which the TREX preprocessor is installed, choose the directory in which the exported root certificate of the Content Management server is to be stored.

You already determined this directory for the environment variable `SECUDIR` on the host on which the TREX preprocessor is installed.

11. Restart the TREX preprocessor if necessary.

Result

You have exported the root certificate of the portal Web server.

Importing the Root Certificate of the Portal Web Server

Use

You import the root certificate of the portal Web server to the keystore `SAPSSLS.pse` that you just created. You do this using the *SAPGENPSE* cryptography tool. Proceed as follows:

Procedure

1. Open a prompt on the server on which the TREX preprocessor is installed.
2. Go to the directory in which the executable file **`sapgenpse.exe`** (Windows) or `sapgenpse` (UNIX) is located.
3. Start the import by *SAPGENPSE* by entering the following:

   `sapgenpse maintain_pk -a <EXPORTED_FILENMAE>.cer -p SAPSSLS.pse`

Overview of Commands for SAPGENPSE

Command	Function
`sapgenpse`	Starts the *SAPGENPSE* cryptography tool.
`maintain_pk`	Function of *SAPGENPSE* that imports the root certificate to the keystore.
`-a <EXPORTED_FILENAME>.cer`	Enter the file name of the root certificate of the portal Web server to be imported. **`<EXPORTED_FILENAME>.cer`** is a placeholder for the exported certificate.
`- p SAPSSLS.pse`	You specify the file name of the keystore that is to contain the root certificate here.

`Access sequence`

Check whether keystores already exist in your SECUDIR directory. As the SAPCRYPTOLIB accesses existing keystores in the order `1. SAPSSLA.pse --> 2. SAPSSLC.pse --> 3. SAPSSLS.pse,` you also have to import the root certificate of the portal Web server to the keystores `SAPSSLA.pse` and `SAPSSLC.pse`. Otherwise you receive an error message.

Result

You have configured anonymous client authentication between the TREX preprocessor and the portal Web server.

See also:

Usage of Keystores [Page 71]

TREX Web Server and TREX Java Client (CM)

Purpose

An element of the TREX software – the TREX Java client – is implemented into Content Management so that Content Management can access the TREX functions. The Java client communicates with the Web server on which a TREX extension is installed. The Web server then forwards requests to the TREX servers.

In most portal applications, the TREX Java client and Web server are behind the firewall. This means that you do not have to configure secure communication between these two components. This depiction relates only to the TREX Web server on Windows, implemented as a Microsoft Internet Information Server (IIS) with a TREX extension. TREX is used by the Apache Web server on UNIX. For information on configuring the Apache Web server for SSL and HTTPS, see SAP Note 620169: *TREX 6.0: SSL and HTTPs for Apache Web Server.*

The Java client and Web server both need a certificate issued by the same CA in order to be able to communicate with one another securely.

- The Java client needs a client certificate.
- The Web server needs a server certificate.
- Both components need the root certificate of the CA that issues the other two certificates.

The two communication partners can then encrypt their messages before sending them. The Web server can also authenticate the Java client using its certificate. The Web server rejects requests from unknown communication partners.

Administrators provide the necessary certificates. They also configure the security settings on the Web server and modify security-relevant parameters in the TREX configuration files.

Prerequisites

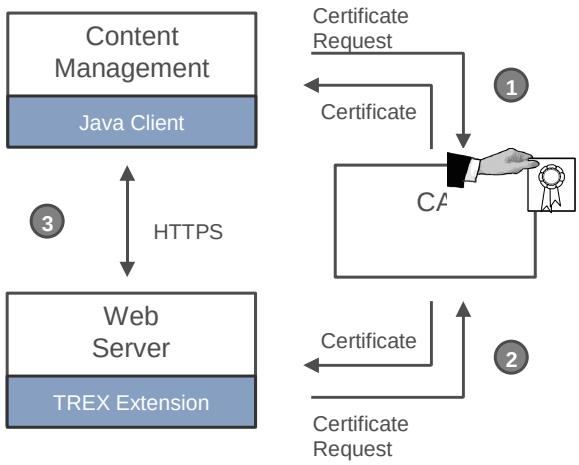
- In your enterprise, you have built up a public key infrastructure with your own CA that issues certificates.

  or

- You are working with any organization that offers the issuing of certificates.

Process Flow

The graphic below gives an overview of administrative tasks.



1. Provide a client certificate and the root certificate of the CA for the Java client.
2. Provide a server certificate and the root certificate of the CA for the web server.
3. The Java client and Web server can communicate using HTTPS.

Providing the Certificates for the Java Client

Purpose

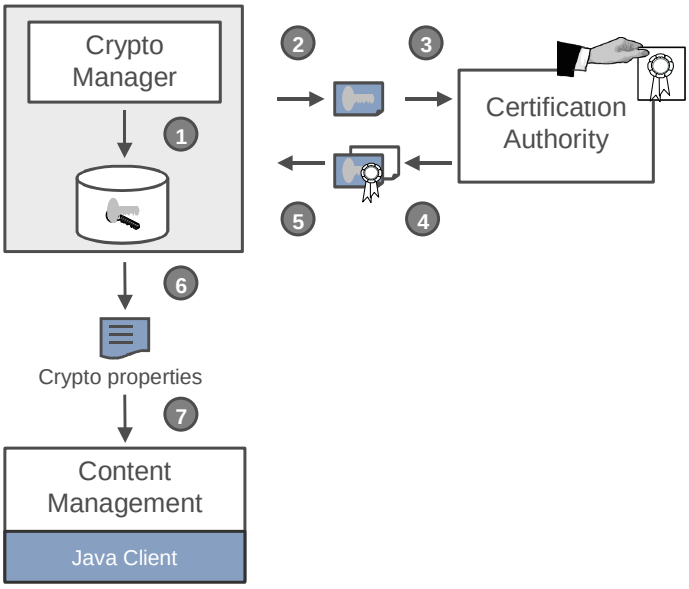
The Java client requires the following so that it can communicate with the Web server using a secure connection:

- A client certificate
- The root certificate of the CA that issues the client certificate

You manage both certificates for the Java client using the SAP Crypto Manager. The Crypto Manager is a tool that you can use to manage the certificates for Java applications.

Process Flow

The graphic below depicts the steps required and the order in which you carry them out.



1. You use the Crypto Manager to create a keystore for the Java client on the portal server. The keystore is a file that contains the public and private key of the certificate owner and that is protected by a password.
2. You now create a certificate request.
3. Send this request to the CA.
4. As soon as the CA has issued the client certificate, you can collect it along with the root certificate of the CA.
5. You import both these certificates to the Crypto Manager.
6. Create an SSL configuration file that contains the path and name for the keystore.
7. You now configure the Java client for SSL. Register the SSL configuration file and the SSL port with the Web server, and define that communication is to take place using HTTPS.

The following sections provide more detailed information on each step.

Creating the Keystore

Use

You use the Crypto Manager to create a keystore for the Java client. The keystore is a file that contains the public and private key of the certificate owner and that is protected by a password.

Procedure

1. Start the Crypto Manager on the portal server.

   The SAP Crypto Manager is part of the portal installation. For more information on calling and using the Crypto Manager, see SAP Note **583 288.**

2. Choose *Create new keystore*.
3. Enter the path and name of the new keystore. Use the … pushbutton next to the entry field to choose a directory for the keystore. If you do not choose a directory, the keystore is created in the Crypto Manager directory.
4. Adopt the *SUN Keystore* format and choose *Next.*
5. Use the fields *Common Name* to *eMail* to enter information that uniquely identifies the owner of the certificate.

Field	Example Entry
Common Name	**myhost.mydomain**
Organizational Unit	**mydepartment**
Organization	**mycompany**
Location	**mycity**
State	**mystate**
Country	**mycountry**
eMail	**myaccount@mydomain**

   Use the *Common Name* entry to enter the path and name of your host and your complete domain. Note that requirements for this entry can differ depending on the certification authority (CA).

6. Adopt the encryption algorithm *RSA* and the key length *1024.* Choose *Next.*
7. Check your entries. To change the entries, choose *Previous*. To finish creating the keystore, choose *Finish.*
8. Enter an alias. The alias is later used in the Crypto Manager as the name of the certificate and the name of the generated key.
9. Enter a password for the keystore. The keystore can only be loaded by users that know this password.
10. Choose *OK*.

    The keystore has now been created, and a key pair consisting of a public and a private key is generated. You receive a message telling you that the process has been completed.

11. Confirm the message with *OK*.

Result

The contents of the keystore are displayed in the main window of the Crypto Manager. You can now generate a certificate request and send it to the certification authority (CA).

Generating the Certificate Request

Use

When you have created a keystore for the Java client, you can generate the certificate request. The certificate request contains:

- Information on the owner of the certificate
- The public key of the owner

You send the certificate request to the certification authority (CA). The CA then issues the actual certificate.

Prerequisites

You have opened the Crypto Manager and loaded the keystore.

Procedure

1. Select the certificate in question under *Contents of certificate* in the Crypto Manager. You identify the certificate by its key symbol.
2. Choose *PKCS#10.*

   You reach a dialog box that contains the certificate request.

    Public key cryptography standards (PKCS) define data formats that are used in public key cryptography. PKCS#10 is a standardized format for a certificate request.
3. Select the entire certificate request using `Ctrl + A,` and copy it to the clipboard using `Ctrl + C.` From there, copy the data to the request form of the CA, or save the data in a text file in order to send the certificate request to the CA.

Result

You can now send the certificate request to the certification authority (CA). The administrator of the CA checks the request and then issues the actual certificate. If you are able to communicate with the CA using the Internet, you can usually check the status of your request, and see whether or not the administrator has processed the certificate yet. You can collect the certificate as soon as the administrator has issued it. You collect the client certificate together with the root certificate of the CA.

Importing Certificates into the Crypto Manager

Prerequisites

You have received a file in PKCS#7 format from the certification authority. It contains the following certificates:

- The certificate for the Java client.
- The root certificate of the CA

Procedure

1. Navigate to the directory that contains file with the certificates.
2. Open the file with the certificates with a text editor.
3. Select all of the content of the file, and copy it to the clipboard using `Ctrl` + `C`.
4. Start the Crypto Manager on the portal server, and load the keystore that you created for the Java client.
5. Select the certificate in question under *Contents of certificate*. You identify the certificate by its key symbol.
6. Choose *Reimport*.
7. Delete the information in the *PKCS#10 Handling* dialog box, and insert the contents of the clipboard using `Ctrl` + `V`.
8. Choose *OK*.

Result

The key store now contains two certificates:

- The certificate generated by you that came back signed.
- The root certificate of the CA

You now generate another SSL configuration file, and configure the Java client.

Generating the SSL Configuration File

Use

After importing the certificate to the crypto manager, you generate an SSL configuration file. The configuration file contains the path and name of the keystore.

Prerequisites

You have opened the Crypto Manager and loaded the keystore that contains the certificates for the Java client.

Procedure

1. In the Crypto Manager, choose the *Properties file* tab.

2. Next to *KeyStore file* and *TrustStore* file, click on  to enter the path and name of the keystore.

   The *KeyStore file* field has to contain the file that contains the certificate for the Java client. The *TrustStore file* field has to contain the file that contains the root certificate of the CA. Since both certificates are stored in the same keystore, both fields should contain the same file.

3. Make sure that *Force client to use certificate* is selected.
4. Choose *Save properties*.
5. Enter the path and name of the configuration file. You can change the proposed name `(crypto.properties)` if necessary.

   For example, `c:\javaprojects\ssl\crypto.properties`

6. Choose *Save* and confirm with *OK.*

Configuring the Java Client for SSL

Use

Now configure the Java client for SSL communication. Register the path of the SSL configuration file, and define that the communication is to take place using HTTPS. Also register the SSL port that the TREX Web server uses.

Prerequisites

You can log on to the portal and have the role *KM Admin.*

Procedure

1. Log on to the portal and choose *KM Admin* → *Configuration* → *TREX* → *TREX Java Client* from the top level navigation bar.
2. Choose *Default HTTP Server.* Edit the *default* entry as follows:

Parameter	Entry
`SSL Configuration File`	Enter the path and name of the SSL configuration file. For example, `c:/javaprojects/ssl/crypto.properties` Only use the forward slash (/) in path specifications.

`HTTP Server`	Modify the protocol and the address of the Web server as follows. **`https`**`://<%trexserver%>:`**`<ssl_port>`**`/TREXHttpServer/TREXHttpServer.dll` The SSL port of the Web server is normally 443. However, if TREX and the portal are installed on the same host, this port is already being used by the portal. If this is the case, choose another SSL port, for example, 444, or any port that is not yet being used. For example, `https://<%trexserver%>:444/TREXHttpServer/TREXHttpServer.dll`  HTTP and HTTPS cannot run on Microsoft IIS on the same port. Therefore, you have to specify a port that is only used for HTTPS communication. The SSL port specified here has to be the same as the SSL port that you use later on when configuring the Web server (see Configuring Secure Communication on the Web Server [Page 91]).  The address can contain the actual hostname of the TREX server, or the specification `<%trexserver%>`. This is a variable for the hostname. The actual hostname is stored on the portal server in the `config_local.properties` configuration file.
`Protocol`	**`https`**
`Search Engine`	**`DRFUZZY`**

3. Choose *HTTP Server.* Edit the *admin.httpserver.0* entry as follows:

Parameter	**Entry**
`Address`	Modify the address of the Web server again. **`https`**`://<%trexserver%>:`**`<ssl_port>`**`/TREXHttpServer/TREXHttpServer.dll`
`Active`	Make sure that this field is selected.
`Access Count`	Accept the default setting.

4. Restart the servlet engine on the portal server.

Result

The Java client is now prepared for secure communication with the Web server.

If you later change from HTTP to secure HTTP communication, you have to change the communication protocol of the existing indexes to HTTPS. Do this in *TREX Configuration* under *KM Admin* → *Configuration* → *TREX* → *TREX Java Client.*

Providing the Certificates for the Web Server

Purpose

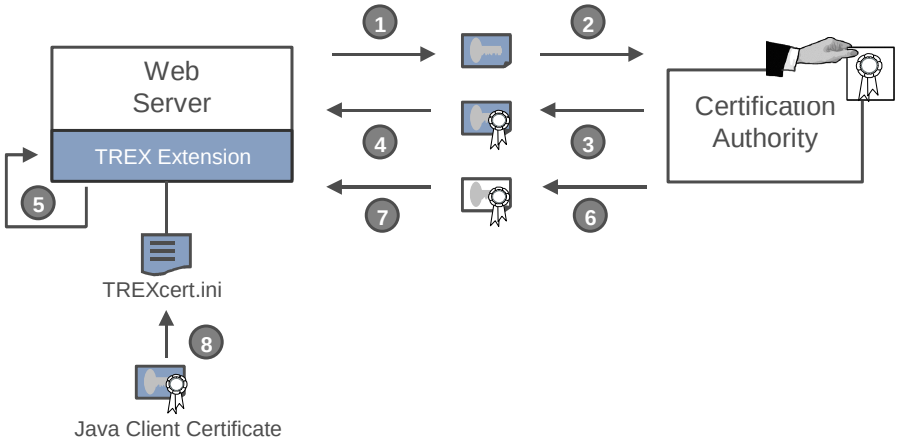
The Web server requires the following so that it can communicate with the Java client using a secure connection:

- A server certificate issued by the same CA as issued the Java client certificate
- The root certificate of the CA

Process Flow

The graphic below depicts the steps required and the order in which you carry them out. For all steps, use the tools provided by Windows and the Web server (Microsoft IIS).

For a distributed TREX installation with multiple Web servers, carry out all steps on each individual Web server.



1. Create a certificate request for the Web server.
2. Send this request to the CA.
3. Collect the certificate for the Web server as soon as the CA has issued it.
4. Import the server certificate to the Web server.
5. Then change the settings for secure communication on the Web server. Define that the communication is to take place using HTTPS. Also define the SSL port to be used for secure communication.
6. Collect the root certificate of the CA.
7. Import the root certificate.
8. Then enter the owner and issuer of the client certificate into the `TREXcert.ini` configuration file. The Web server can authenticate the Java client using the certification information.

The following sections provide more detailed information on each step.

Generating the Certificate Request

Use

The Web server requires a certificate so that it can communicate with the Java client using a secure connection. You firstly have to create a certificate request using the Internet Services Manager.

Procedure

1. Choose *Start* → *Programs* → *Administrative Tools* → *Internet Services Manager*.
2. Use the secondary mouse button to click on the SAP_TREX Web site.
3. Choose *Properties* in the context menu, and then choose the *Directory Security* tab.
4. Choose the *Server Certificate* pushbutton from the *Secure communications* area.

   An assistant starts in order to help you to generate the certificate request.
5. Choose *Next*.
6. Choose *Create a new certificate* and then *Next.*
7. Choose *Create a certificate now, but send it later* and then *Next.*
8. Enter a name for the certificate, for example, **SAPTREX**. Choose 1024 Bit. for the key length.
9. Choose *Next*.
10. Enter the organizational data that distinguishes your organization from another.

Field	Example Entry
Organization	**mycompany**
Organizational Unit	**mydepartment**

11. Choose *Next*.
12. Use the *Common Name* entry to enter the path and name of your host and your complete domain.

Field	Example Entry
Common Name	**myhost.mydomain**

Note that requirements for these entries can differ depending on the certification authority (CA).

13. Choose *Next*.
14. Enter the information on the location of your organization, and then choose *Next.*

Field	Example Entry
Country/Region	**mycountry**
State/province	**mystate**
City/locality	**mycity**

15. Enter the path and name of the file in which the certificate request is stored.
16. Choose *Next*.
17. Check your entries. To change the entries, choose *Back*. To finish generating the certificate request, choose *Next* and then *Finish.*

18. Navigate to the directory that contains the file that contains the certificate request.
19. Open the file with a text editor.
20. Select all of the content of the file using `Ctrl + A`, and copy it to the clipboard using `Ctrl + C`. From there, copy the data to the request form of the CA, or save the data in a text file in order to send the certificate request to the CA.

Result

You can now send the certificate request to the certification authority (CA). The administrator of the CA checks the request and then issues the actual certificate. If you are able to communicate with the CA using the Internet, you can usually check the status of your request, and see whether or not the administrator has processed the certificate yet. You can collect the certificate as soon as the administrator has issued it.

Importing the Certificate to the Web Server

Prerequisites

You have collected a certificate for the Web server from the certification authority (CA).

Procedure

1. Choose *Start* → *Programs* → *Administrative Tools* → *Internet Services Manager*.
2. Use the secondary mouse button to click on the SAP_TREX Web site.
3. Choose *Properties* from the context menu, and then choose the *Directory Security* tab.
4. Choose the *Server Certificate* pushbutton from the *Secure communications* area.

   An assistant starts in order to help you to import the certificate.
5. Choose *Next*.
6. Choose *Process the pending request and install the certificate,* and then choose *Next.*
7. Enter the path and the name of the file that contains the certificate. The file name is `certnew.cer` by default.
8. Choose *Next*.

   The certificate data is displayed.
9. Choose *Next* and then *Finish.*

Result

You can now configure the settings for secure communication on the Web server.

Configuring Secure Communication on the Web Server

Use

After you have imported the certificate for the Web server, configure the settings for secure communication.

Firstly, you define the protocol to be used for communication with the Web server. We recommend that you choose HTTPS communication, since it is the only way of ensuring secure communication and authentication of the communication partner.

In the next step, you specify the SSL port to be used for the HTTPS communication.

Prerequisites

You have imported the certificate for the Web server to the Web server. You have opened the Internet Services Manager and are displaying the properties of the SAP_TREX Web site (*SAP_TREX Properties* dialog box).

Configuring HTTPS Communication

1. Choose the *Directory Security* tab, and then choose the *Edit* pushbutton from the *Secure communications* area.
2. Select the *Require secure channel* field.
3. Select the *Require client certificates* field.
4. Choose *OK*.

Defining the SSL Port

1. Choose the *Web Site* tab.
2. Specify the SSL port.

   The SSL port of the Web server is normally 443. However, if TREX and the portal are installed on the same host, this port is already being used by the portal. If this is the case, choose another SSL port, for example, 444, or any port that is not yet being used.

   HTTP and HTTPS cannot run on Microsoft IIS on the same port. Therefore, you are not allowed to use the same number for both the TCP port and the SSL port.

   The SSL port entered here has to be the same as the SSL port that you entered when configuring the Java client (See Configuring the Java Client for SSL [Page 86]).
3. Choose *OK*.

Result

The SAP_TREX Web site is fully configured. You now need to provide the root certificate of the CA to the Web server.

Importing the Root Certificate of the CA

Use

The Web server needs the root certificate of the certification authority (CA) that issued the certificates for the Web server and the Java client.

Prerequisites

You have collected the root certificate of the CA and stored it in any directory. The procedure for collecting the root certificate depends on the CA in question.

Procedure

1. Navigate to the directory that contains the root certificate of the CA.
2. Double-click on the certificate file `<name>.cert`.
3. Choose *Install Certificate*.

   An assistant starts in order to help you to import the certificate.
4. Choose *Next*.
5. Choose *Place all certificates in the following store,* and then choose *Browse*.
6. Select the *Show physical store* field.
7. Under *Trusted Root Certification Authorities,* select the folder *Local Computer.* This installs the root certificate of the CA centrally for the current host.
8. Choose *OK*.
9. Choose *Next* and then *Finish.*

Result

Now you are ready to make sure that the Web server can authenticate the Java client.

Configuring Authentication

Use

If the Java client sends a request to the Web server during routine operation, it also transmits the public information for its certificate. The Web server uses this information to authenticate the Java client.

The prerequisite for this is that you enter the information from the client certificate into the `TREXCert.ini` configuration file. The Web server compares the information transmitted with the information in the configuration file, and only forwards requests from clients that it recognizes. If the Web server receives a request from a client that it does not recognize, it sends the request back.

You can enter more than one client certificate into the configuration file. This is only beneficial if multiple portals are accessing TREX using secure communication.

For security reasons, you should protect the `TREXCert.ini` configuration file with operating system methods. For example, you can dictate that only certain users can read the file.

The Web server reads the configuration file during routine operation. Therefore, the user on which the IISADMIN service and the WWW publishing service run needs to have read-access to the configuration file.

Prerequisites

You have provided the certificates for the Java client (see Providing the Certificates for the Java Client [Page 81]). To prepare, start the Crypto Manager on the portal server, and load the keystore that contains the certificates for the Java client.

Procedure

1. Open the configuration file `<TREX_Directory>\TREXTCert.ini` on the TREX Web server with a text editor.
2. In the `[WEBSERVERCERTIFICATEnn]` section, replace the entry `nn` with **`1`** when you enter the first client certificate. You can enter as many client certificates are necessary. Number them sequentially.

   `[WEBSERVERCERTIFICATE1]`

   `subject=`

   `issuer=`

3. In the parameters, enter the holder and issuer of the client certificate.

   You can take this information from the Crypto Manager. If you expand the client certificate there, and select the entry *DN of owner* or *DN of issuer*, you will see the relevant information.

   The keystore in the Crypto Manager contains two certificates: The client certificate and the root certificate for the certification authority (CA). Note that you expand the client certificate and not the root certificate of the CA. You can distinguish between the certificates using the *DN of owner* field.

   You then see the information on the owner and issuer as displayed in the Crypto Manager.

   Owner: `CN=myhost.mydomain, OU=mydepartment, O=mycompany, L=mycity, ST=mystate, C=mycountry, EMail=myaccount@mydomain.`

   Issuer: `CN=My Certificate Authority (CA), OU=Certificate Center, O=CA Company, L=CA City, ST=CA State, C=CA Country, EMail=caaccount@cacompany.com.`

   You enter this information into the `TREXCert.ini` configuration file as follows:

   ```
   [WEBSERVERCERTIFICATE1]

   subject=CN=myhost.mydomain, OU=mydepartment, O=mycompany,
   L=mycity, ST=mystate, C=mycountry, EMail=myaccount@mydomain

   issuer=CN=My Certificate Authority (CA), OU=Certificate Center,
   O=CA Company, L=CA City, ST=CA State, C=CA Country,
   EMail=caaccount@ cacompany.com
   ```

   Note the following:

   - **Names** – The holder of the certificate is known as the `owner` in the Crypto Manager and the `subject` in the configuration file.

Several names are permitted in the configuration file for the following information.

Information	Name in TREXCert.ini
State	ST= and S=
E-mail address	EMail= and E=

The other names need to be entered into the configuration file exactly as they appear in the Crypto Manager, that is, `O=` is the organization, `L=` is the location, and so on.

- **Spelling** – The spelling of the entries in the configuration file must be exactly the same as in the Crypto Manager. Pay particular attention to lower- and upper case and to the spacing in parameter values.
- **Order** – The order of the values in the parameters `subject=` and `issuer=` is irrelevant. For example, both `subject=CN=myhost.mydomain, OU=mydepartment, O=mycompany, ...` and `subject=O=mycompany, OU=mydepartment, CN=myhost.mydomain, ...` are permitted.
- **Line breaks** – All values for one parameter (for example, in `subject=`) must be on a single line. Separate entries on the line using commas and spaces as in the example below.

  ```
  issuer=E=caaccount@ cacompany.com, C=CA Country, ...
  ```

4. Save the `TREXCert.ini` file and close the text editor.

Result

If a client that is not entered into the `TREXCert.ini` configuration file sends a request to the Web server, the request is rejected with status 403 (access denied). The Web server also rejects requests if

- No client certificate has been sent
- The client certificate sent is from a CA that the Web server does not trust

See also:

Troubleshooting [Page 94]

Troubleshooting

Use

If the Web server rejects requests from the Java client with the status 403 (access denied), this can be due to incorrect entries in the `TREXCert.ini` configuration file. Below is a description of how to check the determined data using the Windows Event Viewer and correct the `TREXCert.ini` configuration file.

Procedure

1. Open the configuration file `<TREX_Directory>\TREXTCert.ini` on the TREX Web server with a text editor.
2. Set the `tracelevel` parameter in the `[TRACE]` section to **2**.

```
[TRACE]
tracelevel=2
```

3. Save the `TREXCert.ini` file and close the text editor.
4. Restart the Web server.
5. In the portal, repeat the action that failed (start the search or create the index, for example).
6. Start the Event Viewer (*Start* → *Programs* → *Administrative Tools* → *Event Viewer)*.
7. Choose *Application Log*.

   Two events are created for each request. One event contains information for the owner of the client certificate, and the other contains information for the issuer.
8. Use the secondary mouse button to click on the first of the events created by the request that failed. Choose *Properties* from the context menu.
9. In the *Description* field, select all the information from the client certification, without the period/full stop at the end.

   The *Description* field can contain the following text:

   ```
   The description of Event ID ( 1 ) in Source ( SAP TREXHttpServer
   for ISAPI ) cannot be found. The local computer may not have the
   necessary registry information or message DLL files to display
   messages from a remote computer. The following information is
   part of the event: E=myaccount@mydomain, C=mycountry, S=mystate,
   L=mycity, O=mycompany, OU=mydepartment, CN=myhost.mydomain
   ```

   If this is the case, select the following information:

   ```
   E=myaccount@mydomain, C=mycountry, S=mystate, L=mycity,
   O=mycompany, OU=mydepartment, CN=myhost.mydomain
   ```
10. Copy this information.
11. Open the `<TREX_Directory>\TREXCert.ini` configuration file with a text editor.
12. Add the information to the `[WEBSERVERCERTIFICATE1]` section as follows:
    - If you selected the information on the owner, add it after `subject=`.
    - If you selected the information on the issuer, add it after `issuer=`.

    ```
    [WEBSERVERCERTIFICATE1]
    subject=E=myaccount@mydomain, C=mycountry, S=mystate, L=mycity,
    O=mycompany, OU=mydepartment, CN=myhost.mydomain
    issuer=E=caaccount@cacompany.com, C=CA Country, S=CA State,
    L=CA City, O=CA Company, OU=Certificate Center, CN=My Certificate
    Authority (CA)
    ```
13. Repeat steps 8 to 12 for the second event.
14. Save the `TREXCert.ini` file and close the text editor.
15. In the portal, repeat the action that failed again (start the search or create the index, for example).

    The request should now be sent successfully to the Web server. If problems still occur, contact TREX support.

16. For security reasons, set the `tracelevel` parameter in the `TREXCert.ini` configuration file to **`0`** again. Then restart the Web server.

TREX Web Server and TREX ISAPI Register (Windows Only)

Purpose

The Content Management Java client accesses TREX functions using the TREX Web server. Communication between Content Management and the Web server takes place using HTTP or HTTPS and XML. The TREX ISAPI Register component is installed on the Web server. This enhances the Web server with TREX-specific functions. Technically, this component is an ISAPI server extension for the Microsoft© Internet Information Server (IIS). On Windows, the ISAPI Register makes sure that the Web server registers with the name server after starting. The name server is then able to recognize the Web server and forward its address on request. If the TREX Java client and Web server are to communicate using HTTPS protocol, you have to configure the TREX ISAPI Register component for secure communication.

 You only have to configure secure communication between the TREX Web server and the TREX ISAPI register for a TREX installation on Windows. On UNIX, TREX uses the Apache Web server. This registers itself with the TREX name server when it starts, and technically does not have an ISAPI extension.

Prerequisites

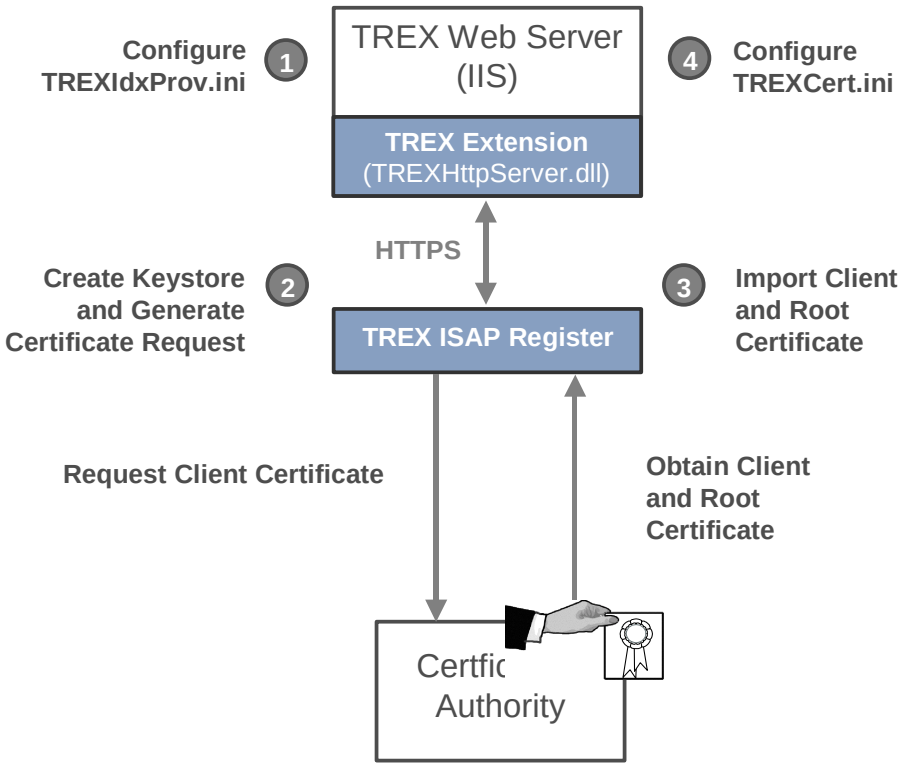
You have installed and configured the SAP cryptography tool *SAPGENPSE* on the host on which the TREX preprocessor is running. *SAPGENPSE* is a part of the *SAP Cryptographic Library*, which is the standard security product delivered by SAP for encryption functions in SAP systems. The *SAP Cryptographic Library* is available as an installation package for download by authorized customers in the SAP Service Marketplace. You can find more information on downloading and configuring *SAPGENPSE* under Usage of SAP Cryptography Tools [Page 64].

Process Flow

You carry out the following steps to configure the *TREX ISAPI Register* component for HTTPS:

1. Configure the INI file TREXIdxProv.ini.
2. Create keystores and request certificates.
3. Import client and root certificates.
4. Configure the INI file TREXCert.ini.

The figure below gives an overview of the process flow.



Configuring the INI file TREXIdxProv.ini.

Use

To prepare the TREX ISAPI Register component for HTTPS, change the communication protocol from HTTP to HTTPS in the configuration file `TREXdxProv.ini` and enter the name of the keystore that you later create using the cryptographic tool *SAPGENPSE.*

We recommend that you call the keystore for the TREX ISAPI Register **`sapISAPIssl.pse.`**

Procedure

1. Using a text editor, open the configuration file `<TREX_Directory>\TREXIdxProv.ini` on the host on which the TREX Web server is installed.
2. In the section `[HTTPServer]`, change the parameter `URL` from `http` to `https:` **`URL=https`**`://<%trexserver%>:444/TREXHttpServer/TREXHttpServer.dll`

   Use port number **444** or another free port number. The default port number `443` is already occupied by the portal.
3. In the same section, for the parameter `certfile`, enter the value **`sapISAPIssl.pse`** for the keystore used.

   `[HTTPSERVER]`

```
URL = https://P54896.wdf.sap-
ag.de:444/TREXHttpServer/TrexHttpServer.dll

certfile = sapISAPIssl.pse
```

4. Save the `TREXCert.ini` file and close the text editor.

Creating Keystores and Requesting Certificates

Use

You use the *SAPGENPSE* cryptography tool to create a request for a client certificate with your certification authority (CA). At the same time, you create a new keystore into which you later import the client and root certificate for your CA.

When creating a keystore, refer to the notes for the usage of keystores [Page 71].

Procedure

1. On the server on which the TREX Web server and the TREX ISAPI Register are installed, use *Start* → *Run:* **`cmd`** to open a prompt on Windows.
2. Go to the directory in which the executable file **`sapgenpse.exe`** is located.
3. Create a keystore **`sapISAPIssl.pse`** and a request for a client certificate with your CA in which you specify the following:

   `sapgenpse gen_pse -p sapISAPIssl.pse`

Overview of Commands for SAPGENPSE

Command	Function
`sapgenpse`	Starts the cryptography tool *SAPGENPSE*
`gen_pse`	Function of *SAPGENPSE* that you can use to generate a new keystore and a certificate request.
`-p sapISAPIssl.pse`	You specify the file name of the keystore that contains the client certificate here. We recommend entering the name `sapISAPIssl.pse` for the keystore.

You are now asked to give more precise specifications on the certificate that you want to generate. Proceed according to the following table:

Prompt	Function/Entry
`Please enter PIN:`	Do **not** enter a value. Confirm with Return.
`Please reenter PIN:`	Do **not** enter a value. Confirm with Return.

`get_pse: Distinguished name of PSE owner:`	Specifies the distinguished name (DN) of the certificate owner. Make the following specifications: **`CN=myhost.mydomain, C=mycountry, S=mystate, O=mycompany, OU=mydepartment`** Example: **`CN=p64883.wdf.sap.corp, C=DE, S=BW, O=SAP-AG, OU=TREX`**

4. After you have created a keystore, you have to initialize it for use. Enter the following command to do this:

   `sapgenpse seclogin -p C:\sec\sapISAPIssl.pse`

Command	Function
`seclogin`	Function of *SAPGENPSE* that you use to initialize a new keystore for use.
`-p C:\sec\ sapISAPIssl.pse`	Specify the path and file name of the keystore that you want to initialize.

   You are now asked to authorize this process by entering a password.

Prompt	Function/Entry
`Please enter PIN:`	Do **not** enter a value. Confirm with Return.
`Please reenter PIN:`	Do **not** enter a value. Confirm with Return.

5. On Windows, you have to give the local system NT user access permission to the keystore files. Otherwise, the operating system cannot access the files. Enter the following command to do this:

   `sapgenpse seclogin -p sapISAPIssl.pse -O SYSTEM`

   Overview of Commands for SAPGENPSE

Command	Function
`-O SYSTEM`	`SYSTEM` gives the local system NT user access to the keystore.

Result

You have created a `sapISAPIssl.pse` keystore **for the storage of certificates** as well as a certificate request that you can now send to your CA. The administrator of the CA checks the request and then issues the actual certificate. You collect the client certificate together with the root certificate of the CA. Next you can import and store the requested client and root certificates for your CA into the keystore `sapISAPIssl.pse` that was created.

Importing Client and Root Certificates

Prerequisites

You have received a file from your certification authority (CA). It contains the following certificates:

- The client certificate for the TREX-ISAPI Register component in the form `PCNAME_client_cert.cer`

  `p54896_client_cert.cer`

- The CA root certificate in the form `CERTIFICATE_AUTHORITY.cer`

  `SAPNetCA.cer`

Procedure

You import the client certificate and the root certificate from your CA with the *SAPGENPSE cryptography tool:*

1. On the server on which the TREX Web server is installed, use *Start* → *Run:* **`cmd`** to open a prompt on Windows.
2. Go to the directory in which the executable file **`sapgenpse.exe`** is located:
3. Import the client and root certificates into the previously-created keystore `sapISAPIssl.pse`, in which you specify the following:

   `sapgenpse maintain_pk -c PCNAME_client_cert.cer –a CERTIFICATE_AUTHORITY.cer -p sapISAPIssl.pse`

   `sapgenpse maintain_pk -c PCNAME_client_cert.cer –a SAPNetCA.cer -p sapISAPIssl.pse`

Overview of Commands for SAPGENPSE

Command	Function
`sapgenpse`	Starts the *SAPGENPSE* cryptography tool.
`maintain_pk`	Imports the response from the CA to a certification request.
`-c PCNAME_client_cert.cer`	File name that contains the client certificate.
`-a CERTIFICATE_AUTHORITY.cer`	File name that contains the root certificate.
`-p sapISAPIssl.pse`	Keystore file name that contains the client certificate. We recommend entering the name `sapISAPIssl.pse` for the keystore.

A SAPSSLS.pse keystore has to exist in the directory that you have defined under the SECUDIR environment variable. However, if no SAPSSLS.pse keystore exists yet, create one or copy the sapISAPIssl.pse keystore, and rename the copy in SAPSSLS.pse. Refer to the notes for the usage of keystores [Page 71].

Result

You have imported the client and root certificates from your CA into the sapISAPIssl.pse keystore.

Configuring the INI file TREXCert.ini.

Use

When the TREX ISAPI Register component sends a request to the TREX Web server, it also transmits the public information for its certificate. The Web server uses this information to authenticate the TREX ISAPI Register. The prerequisite for this is that you enter the information from the TREX ISAPI Register certificate into the TREXCert.ini configuration file. The Web server compares the information transmitted with the information in the configuration file, and only forwards requests from clients that it recognizes. If the Web server receives a request from a client that it does not recognize, it sends the request back. You can enter more than one client certificate into the configuration file.

For security reasons, you should protect the TREXCert.ini configuration file with operating system methods. For example, you can dictate that only certain users can read the file.

Procedure

1. Open the configuration file <TREX_Directory>\TREXIdxProv.ini on the TREX Web server with a text editor.
2. In the [WEBSERVERCERTIFICATEnn] section, replace the entry nn with **1** when you enter the first client certificate. If a client certificate has already been entered, number it respectively. You can enter as many client certificates as you want. You need to number these in ascending order.

   ```
   WEBSERVERCERTIFICATE1]
   subject=
   issuer=
   ```

3. In the parameters, enter the subject= and issuer= of the client certificate.

   You can get this information from the **sapISAPIssl.pse** keystore with the following **SAPGENPSE** command:

   ```
   sapgenpse get_my_name -p sapISAPIssl.pse
   ```

4. You enter the information displayed into the TREXCert.ini configuration file as follows:

   ```
   [WEBSERVERCERTIFICATE1]
   ```

```
subject=CN=myhost.mydomain, OU=mydepartment, O=mycompany, L=mycity,
ST=mystate, C=mycountry, EMail=myaccount@mydomain

issuer=CN=My Certificate Authority (CA), OU=Certificate Center,
O=CA Company, L=CA City, ST=CA State, C=CA Country, EMail=caaccount@
cacompany.com
```

5. Note the following:

   - **Names** – The holder of the certificate is known as the `owner` in the SAPGENPSE cryptography tool and the `subject` in the configuration file.

     The following names are permitted for information below that is entered into the configuration file.

Information	Name in TREXCert.ini
State	`ST=` and `S=`
E-mail address	`EMail=` and `E=`

     The other names need to be entered into the configuration file exactly as they appear in the Crypto Manager, that is, `O=` is the organization, `L=` is the location, and so on.

   - **Spelling** – The spelling of the entries in the configuration file must be exactly the same as in the keystore. Pay particular attention to lower- and upper case and to the spacing in parameter values.

   - **Order** – The order of the values in the parameters `subject=` and `issuer=` is irrelevant. For example, both `subject=CN=myhost.mydomain, OU=mydepartment, O=mycompany, ...` and `subject=O=mycompany, OU=mydepartment, CN=myhost.mydomain, ...` are permitted.

   - **Line breaks** – All values for one parameter (for example, in `subject=`), must be on a single line. Separate entries on the line using commas and spaces as in the example below.

     ```
     issuer=E=caaccount@ cacompany.com, C=CA Country, ...
     ```

6. Save the `TREXCert.ini` file and close the text editor.

Result

You have configured secure communication between the TREX-ISAPI Register and the TREX Web server.

User Management and Security Files

Files used by user management and security components

File description	Path
sapum.properties Contains configuration parameters for user management and security	<SAP_J2EE_Engine_installation_directory>\ume
authschemes.xml Contains definition of authentication schemes available in the portal.	<SAP_J2EE_Engine_installation_directory>\ume
ticketKeyStore File store for key pair.	<SAP_J2EE_Engine_installation_directory>\ume
verify.der Certificate of the Portal Server in DER format.	<SAP_J2EE_Engine_installation_directory>\ume
verify.pse Certificate of the Portal Server in Secude PSE format.	<SAP_J2EE_Engine_installation_directory>\ume

Naming Conventions for Paths in Documentation

The following naming conventions are used throughout this documentation:

Name of Path	Actual Path on Server
<J2EE_Server>	• If you are using an alone version of SAP J2EE Engine: <SAP_J2EE_Engine_installation_directory>/alone • If you are using a clustered version of SAP J2EE Engine: <SAP_J2EE_Engine_installation_directory>/cluster/server

Documentation References

The following table lists external documentation referenced in the Security Guide and indicates where to find the documentation.

Document Name	Location
Configuring the Use of SSL on the SAP J2EE Engine	**http://service.sap.com/security** → *Security in Detail* → *Secure System Management* → *Configuring the Use of SSL on the SAP J2EE Engine.*
Enabling SSL redirection with the ISAPI module	<SAP_J2EE_Engine_installation_directory>\tools\lib\IIS_module\ssl\ISAPI installation guide for SSL support.doc