

Configuring HTTPs Connection in SAP PI 7.10



Applies to:

SAP NetWeaver 7.1x

For more information, visit the [SOA Management homepage](#).

Summary

In the new version on SAP PI there are several changes in Https configuration method that will be resuming in this document.

Authors: Jon Andoni Suarez Moreno, Carlos Iván Prieto Rubio

Company: Realtech System

Created on: 20 October 2009

Author Bio



Carlos Iván Prieto has been working in Realtech System for the last five years as SAP PI Administrator, SAP PI Developer Consultant and SAP J2EE Developer.



Jon Andoni Suarez has been working in Realtech System for the last two years as .Net Developer Consultant, SAP PI Administrator and SAP PI Developer Consultant.

Table of Contents

Introduction	3
Example of SSL execution.....	3
Customizing the SSL service in the ICM	4
Customizing the default profile.....	4
Creating and configuring certificates for SSL communication	5
Create the certificate signed by a Certification Authority	7
Related Content.....	11
Disclaimer and Liability Notice.....	12

Introduction

In order to start a securization of the HTTPs service and allow the exchange of encrypted information through SAP PI, we will follow some simple steps to customize and activate the SSL functionality.

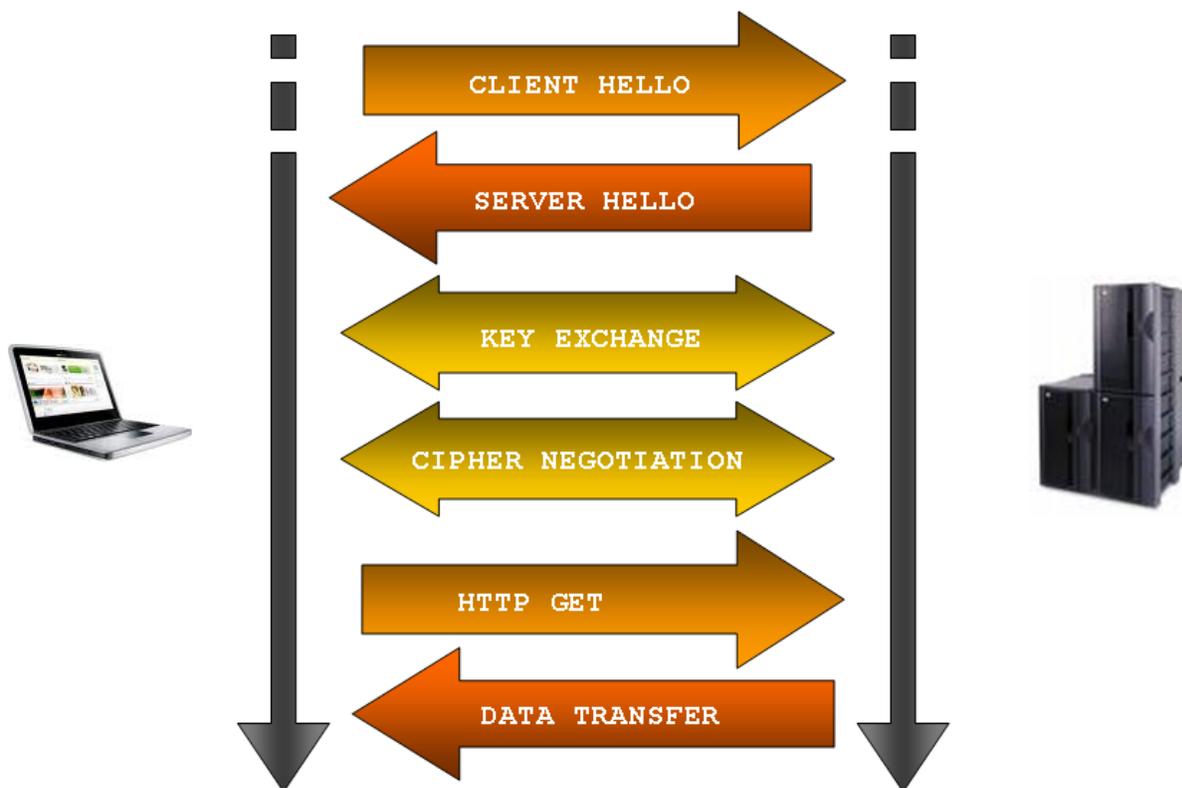
Example of SSL execution

First of all, we think that it's a good exercise remember how works SSL communication. SSL is a protocol used for encrypt the communication over networks like Internet.

For several business scenarios like e-commerce, access to bank data, etc is necessary that communication will be secures over the network.

SSL use asymmetric method for interchange the secret key, this method use a private key and public key. The private key is in server side and the public key is used by client for encrypt or decrypt the messages.

In the next picture is show the mechanism involved between two systems using SSL for securing the communication.



Step 1: The client sends a message to server via Https 1443 port, this is the generic port used by SAP system, more specifically by ICM component. In the first action, the client said "hello" to server.

Step 2: The server response with "hello" if the client is ok and sends the public key, the server certificate, the algorithm to use and one random number. The algorithm to use will be the strongest algorithm that support.

Step 3: The client check if public key isn't expired, then check CN value is the server name and check if one of the CAs installed in client side trust in this public key. The client generate a random key using the server public key and the selected algorithm and sends to server side.

Step 4: In this point, the server and client know the random key, the client generate it and the server receive this key.

Step 5: Send and receive data for both sides.

Customizing the SSL service in the ICM

In old SAP versions there are some methods to configure SSL. One was in ABAP stack and the other one was in J2EE stack. In new version all configuration is made in the ICM component.

This document doesn't show how to configure old SAP versions and is focused in SAP PI 7.1 version.

Once login in SAP system the next steps must to be done:

Go to transaction code SMICM to check the ICM configuration. Information of the configured ports appears following 'Goto' -> 'Services'.

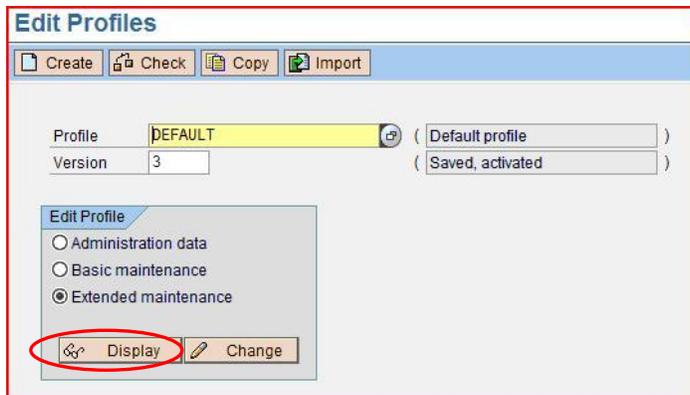
The screenshot shows the 'ICM Monitor - Service Display' window. It contains a table with the following data:

No.	Protocol	Service Name/Port	Host Name	Keep Alive	Proc.Timeo	Actv	External	Bind
1	HTTP	50000	localhost	60	600	✓		
2	P4	50004	localhost	30	60	✓		
3	HTTPS	50010	localhost	30	90	✓		
4	TELNET	50008	localhost	30	60	✓		
5	SMTP	0	localhost	120	120	✓		

Configuration of the ports can be changed by highlighting the desired port and selecting 'Service' -> 'Change'. Check the configuration and activate the service if it is disabled.

Customizing the default profile

Parameters can also be maintained, however, this configuration will be lost after a restarting of the J2EE server, so, if you want to set a permanent configuration, you must change it in the Default Profile of the SAP PI Server. Therefore, go to transaction code RZ10 and select your active DEFAULT profile, as shown in the picture below.



Select the parameter of the list which shows the https configuration, in our example "icm/server_port_2".

Display Profile 'DEFAULT' Version '000003'	
9.10.2009 Active parameters 11:17:59	
Parameter Name	Parameter value
SAPDBHOST	...
j2ee/dbtype	ora
j2ee/dbname	...
j2ee/dbhost	...
SAPSYSTEMNAME	...
SAPGLOBALHOST	...
system/type	DS
rdisp/bufrefmode	sendon,exeauto
DIR_PUT	/usr/sap/\$(SAPSYSTEMNAME)/put
rdisp/mshost	...
rdisp/msserv	...
rdisp/msserv_internal	3900
j2ee/scs/host	...
j2ee/scs/system	10
j2ee/ms/port	3910
login/system_client	001
rdisp/TRACE	1
sec/dsakeylengthdefault	1024
sec/rsakeylengthdefault	1024
icm/HTTPS/verify_client	1
icm/server_port_2	PROT=HTTPS, PORT=1443, TIMEOUT=90
ssl/ssl_11b	/usr/sap/.../SYS/exe/run/11sapcrypto.so

Maintain parameter's values to customize it.

Creating and configuring certificates for SSL communication

The first thing we need is the server certificate signed by a trusted entity.

An unsigned server certificate will be created using the NetWeaver Administrator of the SAP PI:

Log on to NetWeaver Administrator and go to tab 'Configuration Management', 'Security' and 'Certificate and keys'.

Create a 'Key Storage View' under the name ICM_SSL_<instance number> if needed. Once it has been created, you will have a view as shown in the following picture:

The screenshot displays the 'Certificates and Keys: Key Storage' interface. The main table lists several Key Storage Views:

Status	Name	Type	Description
✓	CLIENT_ICM_SSL_47799	USER	ICM Client SSL credentials store
✓	DBMS_User_Store	USER	Contains certificates assigned to users in DBMS user store
✓	DEFAULT	SYSTEM	Public view for common use by all components
✓	ICM_SSL_47799	SYSTEM	ICM Server SSL credentials store
✓	TREXKeyStore	USER	Contains keys and certificates used by the TREX service
✓	TicketKeyStore	SYSTEM	Contains the key-pair to use for issuing logon and assertion tickets, as well as the certificates for all trusted ticket issuing systems
△	TrustedCAs	USER	Template view that contains trusted server certificates
✓	UMKeyStore	USER	Contains a key-pair used by the User Management Engine (UME) provider service
✓	WebServiceSecurity	USER	Web Services Security: Keys and trusted certificates for message security (sign, verify signatures, decrypt)

The 'Key Storage View Details' section for 'ICM_SSL_47799' is currently empty, showing the message: "There are no available entries to display".

It is necessary to save server's private and public key inside the created view. In this example will be shown how to create the certificate with the private key of the server using NetWeaver Administrators functionality.

Certificates and Keys: Key Storage

Content Security

Key Storage Views

Status	Name	Type	Description
✓	CLENT_ICM_SSL_47799	USER	ICM Client SSL credentials store
✓	DBMS_User_Store	USER	Contains certificates assigned to users in DBMS user store
✓	DEFAULT	SYSTEM	Public view for common use by all components
✓	ICM_SSL_47799	SYSTEM	ICM Server SSL credentials store
✓	TREXKeyStore	USER	Contains keys and certificates used by the TREX service
✓	TicketKeyStore	SYSTEM	Contains the key-pair to use for issuing logon and assertion tickets, as well as, the certificates for all trusted ticket issuing systems
✓	TrustedCAs	USER	Template view that contains trusted server certificates
✓	UMESecurity	USER	Contains a key-pair used by the User Management Engine (UME) provider service
✓	WebServiceSecurity	USER	Web Services Security: Keys and trusted certificates for message security (sign, verify signatures, decrypt)

Key Storage View Details

View Entries View Properties

Create Delete Rename Import From View Import From File Export To File Generate CSR Request Import CSR Response

Status	Name	Entry Type	Algorithm	Valid From	Valid To
There are no available entries to display					

Entry Details

Key Storage View Details

View Entries View Properties

Add New Key Storage Entry

Step 1 Step 2 Step 3 Step 4

Entry Settings Subject Properties Sign with Key Pair Preview and Create

Entry Settings

Entry Name: * ssl_server_cert

Algorithm: * RSA

Key Length: * 2048

Valid From: * 19/10/2009

Valid To: * 19/10/2029

Store Certificate:

Cancel Back Next Finish

Step 1: Create the certificate

Key Storage View Details

View Entries View Properties

Add New Key Storage Entry

Step 1 Step 2 Step 3 Step 4

Entry Settings Subject Properties Sign with Key Pair Preview and Create

Subject Properties

Add Remove Move Up Move Down

Name	Object Identifier	Value
countryName *	2.5.4.6	ES
stateOrProvinceName	2.5.4.8	Spain
organizationName	2.5.4.10	SAP AG
localityName	2.5.4.7	Wien
organizationalUnitName	2.5.4.11	Product S
commonName *	2.5.4.3	SSL Server Certificate

Note: Symbol (*) in the subject name denotes value is required

Cancel Back Next Finish

Step 2: The commonName parameter should have the server name value.

Key Storage View Details

View Entries | View Properties

Add New Key Storage Entry

Step 1 | Step 2 | **Step 3** | Step 4

Entry Settings | Subject Properties | **Sign with Key Pair** | Preview and Create

Sign Entry With The Following Key Pair

Select Key Pair

Key	Value

Cancel | Back | Next | Finish

Step 3: If we have a Certificate Authority installed we can sign the certificate.

After clicking on 'Finish' in the Preview step we will see how the certificate PKCS is added to our 'Key Storage View'. This certificate contains both private and public key needed for the encryption and decryption processes.

Create the certificate signed by a Certification Authority

In this example we will show how to create the server certificate using the certification authority provided by SAP.

Step 1: Logging in the Support Portal of the SAP Marketplace, it is possible to apply for a free test certificate valid for eight weeks signed by the SAPServerCA.

This service can be reached going to tab 'Maintenance&Service', then 'SAP Trust Center Services' and 'SSL Test Server Certificates'.

SAP SUPPORT PORTAL | Welcome, Carlos Ivan Prieto Rubio

my Profile | my Inbox | my Favorites

HOME | Help & Support | Downloads | Keys & Requests | Data Administration | **Maintenance & Services** | Application Life-Cycle Management | Release & Upgrade Info | Knowledge Exchange

SAP Services Portfolio | SAP Support Offerings | SAP Maintenance Strategy | Business Objects Product Lifecycle | **SAP Trust Center Services** | SAP Hosting Services | SAP Service Catalog

You are here: **SSL Test Server Certificates**

SSL TEST SERVER CERTIFICATES

SAP TRUST CENTER SERVICES

SSL server certificates ensure

- Secure and confidential data transmission
- Your Internet servers belongs to your company

SSL server certificates also ensure that data exchange **within** your company is secure.

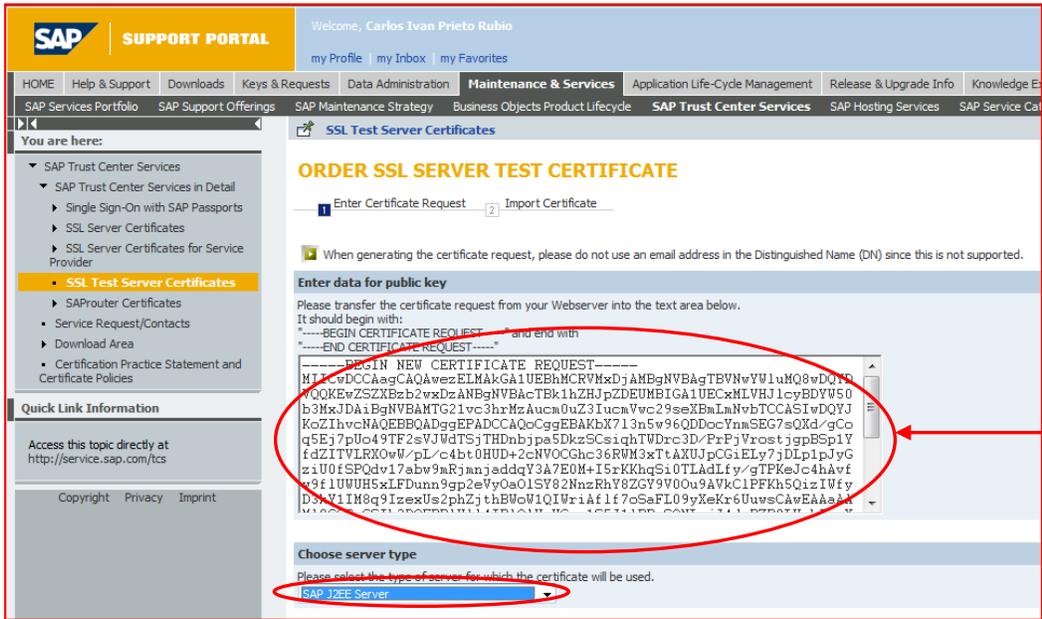
Encrypt your data transfer

SAP Trust Center Services issue SSL Test Server Certificates for any server to enable secure data exchange from web server to browsers (using https protocol) and Client Authentication (using X.509 Client Certificates, for example SAP Passports).

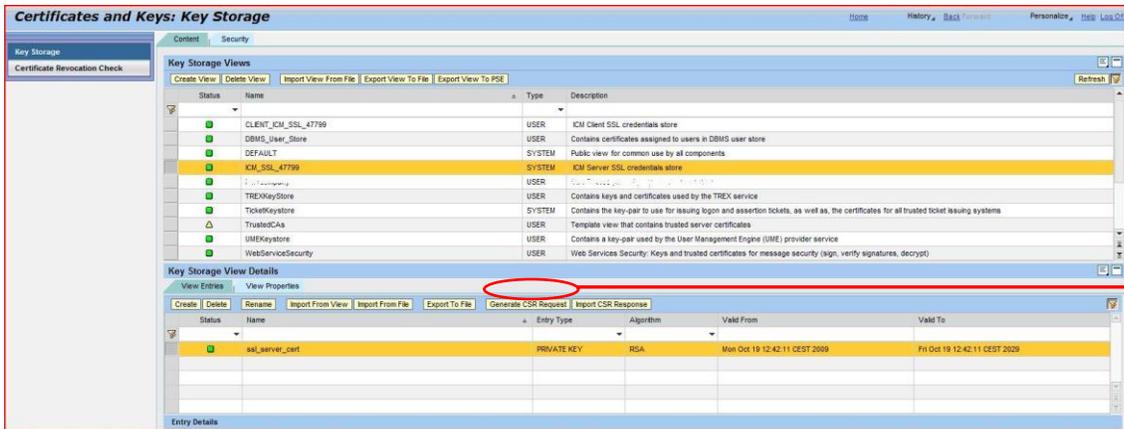
Apply for a free test certificate

On the SAP Service Marketplace, the SSL Test Server Certificates are available free of charge for SAP customers.

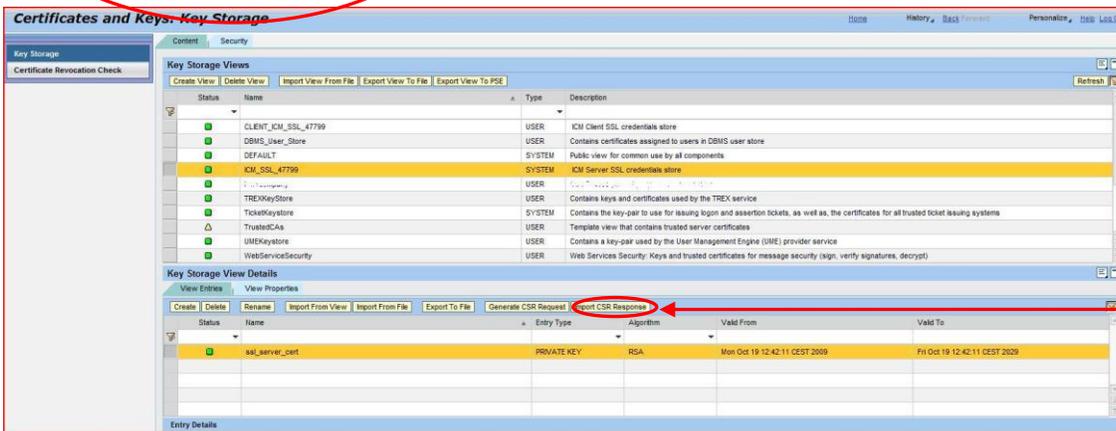
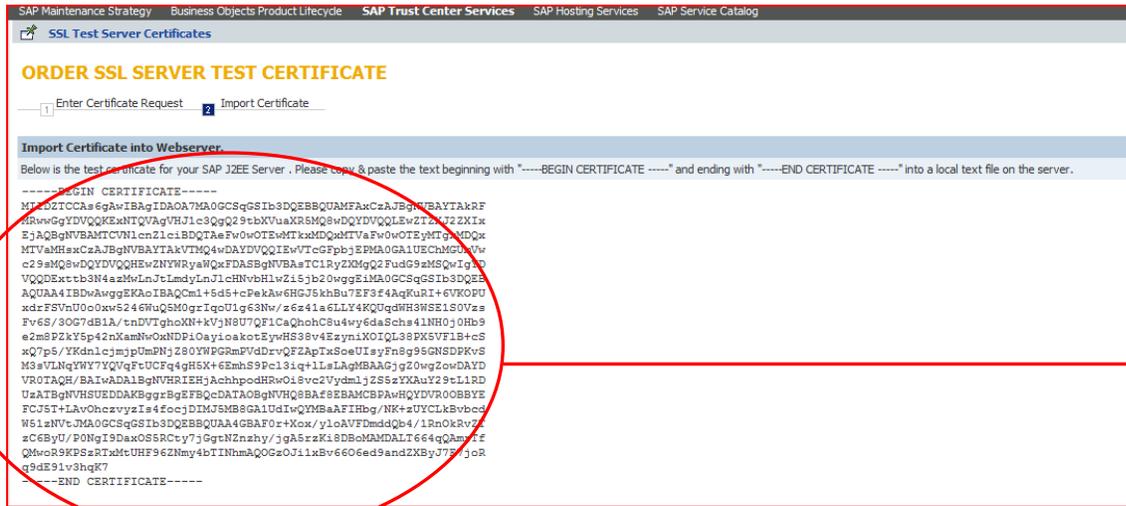
Want to learn more? Visit these related topics:
[SAP Web Application Server](#)



The Certificate Request is obtained from the NWA. Highlight our private key certificate and click on 'Generate CSR Request'. Download the generated file and open it using any word processing software, i.e. Notepad. Copy and paste the content of the file in the web form.

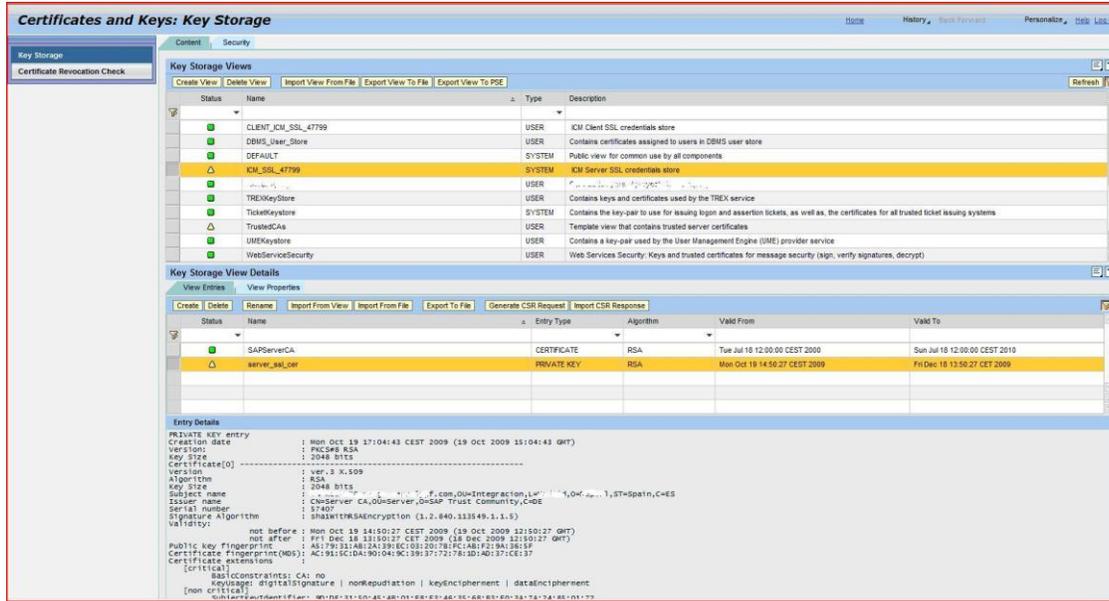


The result will be a signed certificate. The following step is to update the unsigned certificate of our 'Key Storage View' with the information of the signed one. To do so, push on 'Import CSR Response' and copy and paste the signed certificate's string.

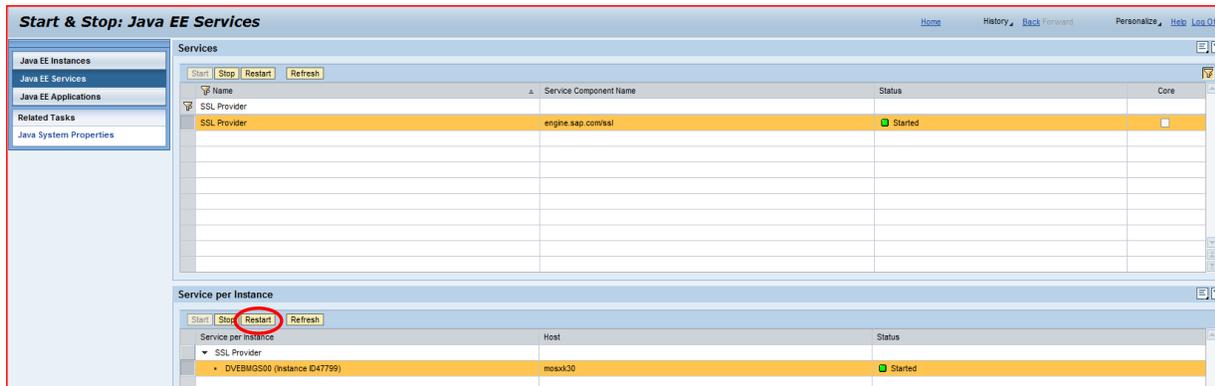


In addition, it is going to be necessary to download the SAP Server CA certificate that is located in 'Maintenance & Services', 'SAP Trust Center Services', 'Download Area', 'Root Certificates' and click over 'SAP Server CA Certificate', in the Support Portal of the SAP Marketplace.

Now, the only step left is to import CA's certificate in our 'Key Storage View' and our web browser using the recently downloaded file. The result of the NetWeaver Administrator is shown in the following screenshot:



Finally, it is necessary to restart the ICM and the keystore service to apply and accept the changes in the keystore. Go to transaction code SMICM and select 'Administration', 'Local ICM', 'Restart' and click on 'Yes'. It is possible to restart keystore service in the NetWeaver Administrator by following the next instructions: go to 'Operation Management', 'Systems', 'Start&Stop'; select 'Java EE Services', search for 'SSL Provider' service and restart it.



Now the SSL securization is switched on our PI system and it is possible to enable it using the URL: https://<server_name>:<SSL_port_configured_in_ICM>

Related Content

[SAP PI 7.1 Security Guide](#)

[Introduction to SSL](#)

[SDN Articles Security](#)

For more information, visit the [SOA Management homepage](#).

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.