Secure Store & Forward / Digital Signatures (BC-SEC-SSF)

HELP.BCSECDISI

Release 4.6C

Copyright

Icons

Icon	Meaning
⚠	Caution
💬	Example
➡	Note
🧭	Recommendation
SYN	Syntax
💡	Tip

Contents

Secure Store & Forward / Digital Signatures (BC-SEC-SSF)

Purpose

Secure Store and Forward (SSF) [Ext.] mechanisms provide you with the means to secure data and documents in SAP Systems as independent data units. By using SSF functions, you can "wrap" data and digital documents in secure formats before they are saved on data carriers or transmitted over (possibly) insecure communication links. The data must not remain within the SAP System; if you save the data in a secure format in the SAP System, it remains in its secured format even if you export it out of the system.

SSF mechanisms use digital signatures [Page 9] and digital envelopes [Page 9] to secure digital documents. The **digital signature** uniquely identifies the signer, is not forgeable, and protects the integrity of the data. Any changes in the data after being signed result in an invalid digital signature for the altered data. The **digital envelope** makes sure that the contents of data are only visible to the intended recipient(s).

The SSF mechanisms are useful in those application areas where an increased level of security exists pertaining to:

- The specific and unique identification of persons or components (for example, in work flow processes)
- Non-repudiation or proof of obligation (for example, when signing paperless contracts)
- Authenticity and integrity of data (for example, saving audit logs)
- The sending or storing of confidential data

By using the SSF mechanisms in SAP applications, you can replace paper documents and handwritten signatures with automated work flow processes and digital documents that are secured with digital signatures and digital envelopes.

Implementation Considerations

SSF mechanisms are available in SAP Systems as of Release 4.0.

You use the SSF mechanisms if you are using an application in the SAP System that has implemented digital signatures or digital envelopes.

There are a number of applications that currently use the SSF mechanisms to provide data protection, for example:

- Production Planning - Process Industry
- Product Data Management
- SAP ArchiveLink - SAP content server HTTP interface 4.5

With time, more and more applications will use SSF for their security purposes.

Secure Store & Forward / Digital Signatures (BC-SEC-SSF)

Constraints

Third-Party Security Product

SSF requires the use of a third-party security product to provide its functions. As the default provider, we deliver the SAP Security Library (SAPSECULIB) [Page 22] with SAP Systems. The SAPSECULIB, however, is limited to providing digital signatures only. For digital envelopes, encryption, or crypto hardware (for example, smart cards or crypto boxes), you need to use a SAP-certified external security product. For a product to be certified by SAP, it must support the PKCS#7 standard data format. For information about supported products, see the SAP Complementary Software Program (http://www.sap.com/csp).

Public-Key Infrastructure

To effectively use the SSF mechanisms, you need to have an established public-key infrastructure (PKI) [Page 9]. The PKI makes sure that you can validate and trust the digital signatures, certificates, and Certification Authorities (CAs). A PKI is often, although not necessarily, supported by the external security products that are available on the market. Although SAP Systems do not provide a PKI directly, they do support PKIs provided by various security products.

Depending on the security product that you use, you can establish the use of a PKI in one of many ways. You may want to create your own PKI and CA that you link to your customers, or you and your customers may want to agree on a common Trust Center. A common Trust Center is a third-party instance that both you and your customers can trust to validate and authenticate your PKI participants. Using a common Trust Center can solve many of the currently open questions regarding the establishment of a PKI.

Laws and Regulations

There are also laws in various countries that regulate the use of cryptography and digital signatures. These laws are currently controversial and may change. You need to keep yourself informed on the impact these laws may have on your applications, and make sure that you are aware of any further developments.

Examples of SAP Applications That Use the SSF Functions

The following SAP applications are examples of areas that use digital signatures to meet their requirements:

- **Quality Management**
  - When saving inspection results for an inspection lot
  - When making and changing the usage decision for an inspection lot
- **Production Planning for Process Industries**
  - When completing a work step in the process industries sheet
  - When accepting invalid values within input validations
  - When approving a batch record
- **SAP ArchiveLink Content Server HTTP interface 4.5**
  - When authenticating a request to access the archive

Secure Store & Forward / Digital Signatures (BC-SEC-SSF)

System Infrastructure for Using SSF Functions

SSF uses an external security product to execute the functions for using digital signatures and encryption in SAP Systems. To communicate with the external security product, the SAP System needs to be able to access the product and its information. Therefore, the following system infrastructure is necessary for using the SSF functions:

- **Communication interface to the security product:**

  To communicate with the external security product on the front ends, the SAP System uses the SSF Remote Function Call (RFC) server program `ssfrfc.exe`. On the application server, the communication interface is included in the SAP System kernel.

- **Access to product-specific information:**

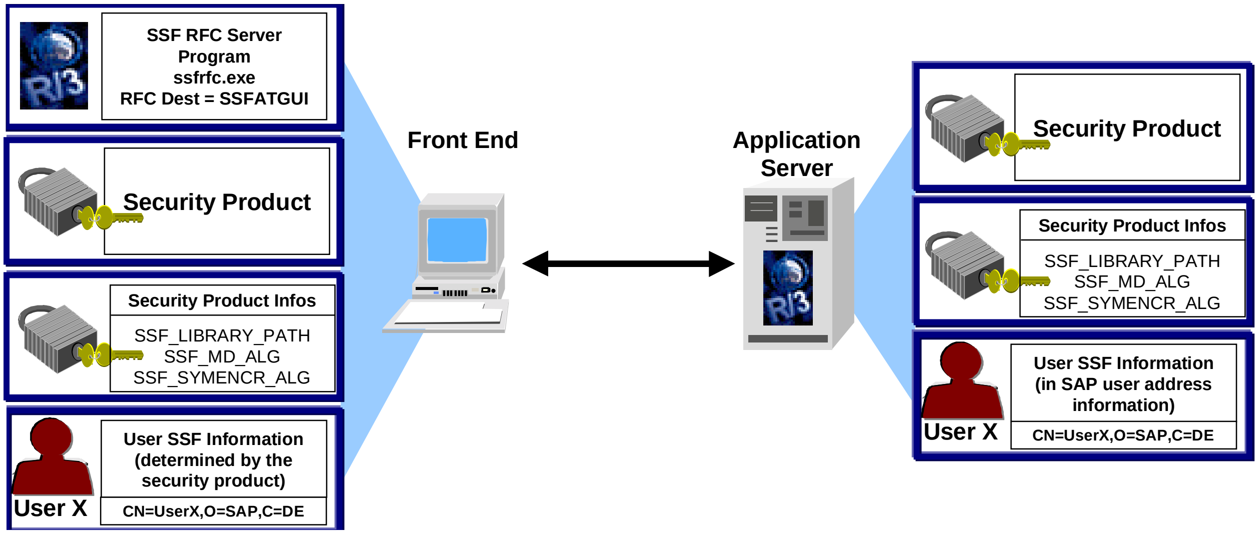
  The SAP System must also be able to access the product-specific information (for example, the location of the library and the algorithms that the product uses). On the front ends, this information needs to be located in either environment variables or in the SSF initialization file `ssfrfc.ini`. On the application servers, this information is contained in either profile parameters or environment variables.

- **User SSF information:**

  In addition, the users that participate in the public-key infrastructure must also be correctly maintained in the SAP System.

See the diagram below.

Components in the System Infrastructure for using SSF



In the following topics, we describe the administration tasks necessary to establish this infrastructure.

Terminology and Abbreviations

Before performing the SSF administration tasks, you should be familiar with the following terms and abbreviations:

- **Certification Authority (CA)**

  A third-party instance that issues public-key certificates. The CA guarantees the identity of the certificate owner.

- **Credentials**

  User or component-specific information that allows users or components to access their security information. The credentials may be located, for example, in a protected file in the file system. They often have a limited life span. For example, users' credentials may be created when they log on to the security product and deleted when they log off.

- **Digital signature**

  Security mechanism for protecting digital data.

  The digital signature serves the same function for the processing of digital data as a handwritten signature serves for paper documents. Its purpose is to guarantee that the individuals (or components) that sign digital documents really are who they claim to be. It also protects the integrity of signed data; if even one bit in either the signed data or in the signature is changed, the signature is invalid.

  The digital signature is based on public-key cryptography. Each signer is provided with a unique key pair consisting of a private key and a corresponding public key. The signer creates his or her digital signature by using his or her private key. He or she distributes the public key as desired. Recipients of signed data use the signer's public key to verify his or her digital signatures.

  For example, in electronic commerce, paperless contracts are closed without using handwritten signatures.

- **Digital envelope**

  Type of security that protects a message from being viewed by anyone other than the intended recipient(s).

  A digital envelope is created using hybrid encryption. First, the message itself is encrypted using symmetric encryption (meaning that the same key is used to encrypt and decrypt the message). This key is then encrypted using public-key encryption and sent or saved with the encrypted message. Only the intended recipient of the message can decrypt the key that was used to encrypt the original message, and therefore, decrypt the message.

- **Personal Security Environment (PSE)**

  Secure location where a user or component's public-key information is stored. The PSE for a user or component is typically located in a protected directory in the file system or on a smart card. It contains both the public information (public-key certificate and private address book) as well as the private information (private key) for its owner. Therefore, only the owner of the information should be able to access his or her PSE.

  For example, the **SAP Security Library** (SAPSECULIB) stores the application server's information in a PSE. In this case, the PSE contains both the **private address book** for the SAP System as well as the **SSF profile**.

- **Private address book**

  Location in the public-key infrastructure where the users’ and components' public keys are stored. Depending on the security product that you use, it may be identical to the SSF profile.

- **Public-key infrastructure (PKI)**

  A system that manages the trust relationships involved with using public-key technology. The PKI's role is to make sure that public-key certificates and CAs can be validated and trusted. The collection of services and components involved with establishing and maintaining these trust relationships is known as the PKI.

- **Public-key technology**

  Technology used for securing digital documents.

  Public-key technology uses key pairs to provide its protection. Each participant receives an individual key pair consisting of a public key and a private key. These keys have the following characteristics:

  – The keys are pairs; they belong together.

  – You cannot obtain the private key from the public key.

  – As the name suggests, the public key is to be made public. The owner of the keys distributes the public key as necessary. A recipient of a signed document needs to have knowledge of this key in order to verify the digital signature. In addition, to send an encrypted document (digital envelope), the sender needs to know the recipient's public key.

  – The private key is to be kept secret. The owner of the keys uses the private key to generate his or her digital signature and to decrypt messages protected with a digital envelope. Therefore, the owner of the keys needs to make sure that **no** unauthorized person has access to his or her private key.

- **Public-key certificate**

  A digital document that contains the necessary information to identify its owner and verify his or her digital signatures. Typical information contained in a public-key certificate include:

  – General information:

    - Version
    - Serial number
    - Validity period

  – Certificate issuer's information:

    - CA's Distinguished Name

  – Certificate owner's information:

    - Owner's Distinguished Name
    - Owner's public key
    - Asymmetric, cryptographic algorithm used

  – CA's digital signature:

  - Asymmetric, cryptographic algorithm used
  - CA's digital signature

- **SAP Security Library (SAPSECULIB)**

  Default security provider provided with the SAP System. The SAPSECULIB is a dynamic link library that is located on each application server. The SAPSECULIB provides the functions for using digital signatures in SAP Systems. It does not support functions for using digital envelopes and encryption.

- **SSF Profile**

  Information in the SAP System where a user or component's private part of the public-key information is stored (the private key). The SSF profile may be a file or any other information specifying the public-key information. The exact form of the profile depends on the security product that you use.

- **System PSE**

  The Personal Security Environment (PSE) for the SAP System. The system PSE is created by the SAPSECULIB during the installation process and contains the private address book and the SSF profile for the SAP System. In Release 4.5A, each application server receives its own system PSE; as of Release 4.5B, the system creates a single system PSE and distributes it to all of the application servers.

See also:

SAP Library:

- BC - Security → Secure Store & Forward / Digital Signatures → Public-Key Technology [Ext.]

SSF Administration Tasks

There are certain administration tasks involved when using the SSF functions. For example, if you use an external security product, you need to install the product on all of the components where the SSF information is needed. You also need to maintain the users who are to use the digital signatures and digital envelopes.

The topics in this section describe the administrative tasks involved with using SSF in SAP Systems. Administrative and maintenance tasks that apply to the SAPSECULIB are also included. For information about tasks that apply to an external security product, see the security product's documentation.

See the following:

- For the standard installation tasks when using an external security product, see the topics under Using SSF with an External Security Product [Page 13]:
  - Installing/Configuring SSF: Front Ends [Page 14]
  - Installing/Configuring SSF: Application Server [Page 15]
  - Maintaining User SSF Information [Page 17]
- If you have maintained user SSF information in Release 4.0 or 4.5 and have upgraded to a later release, see the section Upgrading User SSF Information from Release 4.0/4.5 [Page 20].
- If you use different security products for different applications, see the sections Defining Default SSF Information for Applications [Page 27] and Maintaining Application-Specific Information [Page 28].
- If you use the default security provider SAPSECULIB (digital signatures only), you do not need to perform any installation and configuration tasks. For information about the SAPSECULIB, see the section Using the Default SSF Security Provider SAPSECULIB [Page 21].
- To test the SSF installation, see Testing the SSF Installation [Page 30].

Using SSF with an External Security Product

When using an external security product for providing digital signature and encryption support in SAP Systems, you need to install and configure SSF on each of the frontend computers (see Installing/Configuring SSF: Front Ends [Page 14]) and on the application servers (see Installing/Configuring SSF: Application Server [Page 15]). On the front ends, you can specify the SSF parameter values either in environment variables or in the SSF initialization file `ssfrfc.ini`. On the application servers, you can use either profile parameters or environment variables.

You can only use environment variables to define the SSF parameters as of Release 4.5.

You also need to maintain user SSF information [Page 17] on the application server.

To test your SSF installation, see Testing the SSF Installation [Page 30].

Installing/Configuring SSF: Front Ends

1. Install the security product on each frontend computer where SSF functions are to be used. Note the name and location of the security product's library.

   For installation instructions, see the external security product's documentation.

2. If you want to change the default values of the following SSF parameters, then specify their values either in the corresponding environment variable (as of Release 4.5) or in the file `ssfrfc.ini`. For information about using this file, see the topic The SSF Initialization File [Page 39].

   The following table shows the SSF parameters.

SSF Parameters

Parameter	Default	Possible Values
SSF_LIBRARY_PATH [Page 34]	The default library is the SAPSECULIB library `libssfso.<ext>`. As default, the system searches for this file in the directory where the executable program `ssfrfc.exe` is located.	Character string up to 255 characters. Refer to your security product to find out the name and location of this file.
SSF_MD_ALG [Page 35]	MD5	MD2, MD4, MD5, SHA1, RIPEMD160 Refer to your security product for other possible values.
SSF_SYMENCR_ALG [Page 36]	DES-CBC	DES-CBC, TRIPLE-DES, IDEA Refer to your security product for other possible values.
SSF_TRACE_LEVEL [Page 37]	0	0, 1, 2, 3

Note that the parameter `SSF_LIBRARY_PATH` must contain both the path and the file name of the SSF library.

Installing/Configuring SSF: Application Server

1. Install the security product on each application server. Note the name and location of the security product's library.

   Refer to the external security product's documentation for the installation instructions.

2. Specify the SSF parameters on the application server. You can specify them either in the profile parameters `ssf<x>/...` or in the environment variables `SSF<x>_...` (as of Release 4.5).

   As of Release 4.5B, you can install up to three different security products. This may be necessary if different applications use different security products. Therefore, each product uses its own profile parameter set. Define the parameters for the number of security products that you use. See also Maintaining Application-Specific Information [Page 28].

   The table below shows the application server profile parameters.

SSF Profile Parameters

Parameter	Default	Possible Values
Product 1: `ssf/ssfapi_lib` Product 2: `ssf2/ssfapi_lib` Product 3: `ssf3/ssfapi_lib`	Blank - meaning that the system uses the SAPSECULIB. (See the note below.)	Character string up to 255 characters. Refer to your security product to find out the name and location of this file.
Product 1: `ssf/ssf_md_alg` Product 2: `ssf2/ssg_md_alg` Product 3: `ssf3/ssg_md_alg`	MD5	MD2, MD4, MD5, SHA1, RIPEMD160 Refer to your security product for other possible values.
Product 1: `ssf_symencr_alg` Product 2: `ssf2_symencr_alg` Product 3: `ssf3_symencr_alg`	DES-CBC	DES-CBC, TRIPLE-DES, IDEA Refer to your security product for other possible values.
Product 1: `ssf/name` Product 2: `ssf2/name` Product 3: `ssf3/name`	Product 1: SSF Product 2: SSF2 Product 3: SSF3	Character string up to 10 characters (case-sensitive).

When an application server is started, the system always loads the security product SAPSECULIB and assigns its information to the next available `ssf<x>/...` parameter set.

As default, the system searches for the SAPSECULIB library (`libssfso`) in the directory specified by the profile parameter `DIR_LIBRARY`. If necessary, you can specify a different file name and location of the SAPSECULIB library in the corresponding `ssf<x>/ssfapi_lib` parameter (or in the environment variable `SSF<x>_LIBRARY_PATH`). If you do, make sure that the corresponding parameter `ssf<x>/name` (or the environment variable `SSF<x>_NAME`) contains the name SAPSECULIB.

3. To record SSF activites for trace functions, set the `SSF_TRACE_LEVEL` environment variable to one of the following values:

SSF Trace Levels

Trace level	The system records:
0	• The starting of the SSF RFC server • The loading of the SSF library • The installation of the RFC-enabled SSF functions
1	• Level 0 trace information • The name and return code of SSF functions that are called
2	• Level 0 and 1 trace information • Signer and receiver information when SSF functions are called
3	• Level 0, 1, and 2 trace information • All input and output data when SSF functions are called

The system records the trace information in the file `dev_ssf<#>` (where # is a number assigned to each trace file).

4. Perform any application-specific tasks that may be required. For more information, see the application's documentation.

Maintaining User SSF Information

Depending on your release, you need to maintain user SSF information in different ways.

- **Release 4.0/4.5**

  In Release 4.0, you need to use the Customizing activity *Maintain SSF-Information for the User* (transaction O07C) to enter the user SSF information.

  See also:

  – Information Specific to Release 4.0/4.5 [Page 40]

- **Release 4.6 and higher**

  As of Release 4.6, you maintain the information in the standard user address information.

  See:

  – Maintaining User SSF Information: Release 4.6+ [Page 18]

- **Upgrading from Release 4.0/4.5**

  If you have upgraded to Release 4.6 and have previously maintained user SSF information using Transaction O07C, you need to use the Customizing activity *Upgrade User SSF Information from Release 4.0/4.5* to migrate the SSF information to the user address information.

  See also:

  – Upgrading User SSF Information from Release 4.0/4.5 [Page 20]

Maintaining User SSF Information: Release 4.6+

Use

As of Release 4.6, the user SSF information is included in the standard user address information.

To maintain the user SSF information, use the appropriate transaction as shown in the following table:

Maintenance Transactions for User Address Information

Transaction	Person Responsible	Task
SU01	System administrator	Set up and maintain user master records for all users, to include the users' SSF information.
SO12	Office administrator	Maintain users' office information, to include the users' SSF information for using digital signatures.
SU3	User	Maintain his or her own address information, to include SSF information for receiving encrypted data.

Prerequisites

The security product has been installed and SSF has been configured on the application server (see Installing/Configuring SSF: Application Server [Page 15]).

The location of the SSF RFC server program `ssfrfc` also needs to be defined as the RFC destination `SAP_SSFATGUI` in transaction SM59.

Procedure

1. Call one of the user address maintenance transactions.
2. In the user address information, choose *Other communication*.
3. Select the *Communication type* **SSF**.
4. Enter the following SSF information in the appropriate fields:

User SSF Information

Field	Available with Transaction	Description	Comment		
	SU01	SO12	SU3		
SSF-ID	X	X	X	SSF user name	The syntax of the SSF user name is determined by the security product that you use.

Profile	X		X	Profile name	Specifies the profile where the user's security information is stored. The value of this parameter is also determined by the security product that you use.
Destination	X		X	The RFC destination (logical destination) where the SSF RFC server program has been defined.	See the input help (F4). Default = SAP_SSFATGUI (SSF for digital signatures on the front ends)

You can only maintain those fields that are available in the specific maintenance transaction (see the column *Available with Transaction* in the table above). Fields that are not applicable are not displayed. For example, the office administrator, who uses transaction SO12, cannot change the individual SSF profile for a user.

5. Save the data.

Upgrading User SSF Information from Release 4.0/4.5

In Release 4.6, we moved the user SSF information maintenance to the standard user address maintenance. In Releases 4.0 and 4.5, the user SSF information was stored in table TC70, and as of Release 4.6, the information is stored in table ADR11.

If you have maintained user SSF information in either Release 4.0 or 4.5 and need to upgrade to a later release, then see the Customizing activity *Basis Components* → *System Administration* → *Digital Signatures* → *Upgrade User SSF Information from Release 4.0/4.5.*

This activity runs the report RSADRTC70TOADR11, which moves the user SSF information from table TC70 to ADR11.

Note that this activity is client independent. It transfers the SSF information for users in all of the SAP System clients.

Using the Default SSF Security Provider SAPSECULIB

We deliver SAP Systems with the default SSF security provider SAPSECULIB. The SAPSECULIB provides functions for creating and verifying digital signatures used within the SAP System. It does not provide support for digital envelopes, smart card authentication, or crypto hardware. For complete SSF support, you need to use an external security product.

Normally, you do not have to perform any administrative tasks when using the SAPSECULIB as the security provider. However, the information contained in the following topics may be helpful in the case of problems or when monitoring the status of the SAPSECULIB components.

The SAP Security Library (SAPSECULIB)

Definition

The SAP Security Library (SAPSECULIB) is the default security provider for the SSF mechanisms.

Use

The SAPSECULIB provides the functions for creating and verifying digital signatures within SAP Systems.

Integration

The SAPSECULIB is included as part of the standard SAP System installation. During the installation process, the system uses the SAPSECULIB to generate a Personal Security Environment (PSE) [Page 9] for each application server, called the system PSE [Page 9]. The application server can then use the information contained in the PSE to digitally sign documents and verify other components' digital signatures.

> ➡
>
> In Release 4.5A, the system generates an individual system PSE for each application server.
>
> As of Release 4.5B, the system generates a single system PSE and distributes it to all of the application servers.

The system PSE is created during the installation process and located in the following file in the directory `<instance directory>/sec`:

- Release 4.5A: `SAPSECU.pse`
- As of Release 4.5B: `SAPSYS.pse`

> ➡
>
> When you upgrade from Release 4.5A to a later release, the system creates a new system PSE with the name `SAPSYS.pse`, but does not remove or rename the file `SAPSECU.pse`. Keep in mind that the system may need access to the old PSE to verify digital signatures that were created before the upgrade.

Each time an application server is restarted, the system automatically makes sure that the subdirectory `sec` exists and contains the system PSE for the server. In Release 4.5, if no system PSE is found at system start, then the system automatically generates a new one. As of Release 4.6, if a system PSE exists, then the system distributes the system PSE to the application server. If no system PSE exists in the database, then the system generates a new one for use by all of the application servers.

If you need to generate a new PSE for an application server after the installation process has already been completed, see the topic Maintaining the System PSE [Page 24].

> ➡
>
> **UNIX platforms only:**

So that the system can correctly load the SAP Security Library at application server startup, make sure that the UNIX environment variable for loading shared libraries contains the path referenced by the SAP System profile parameter `DIR_LIBRARY` (for example, `/usr/sap/<SID>/SYS/exe/run`). Make sure the environment variable is set in the user environment for the user account under which the application server runs (for example, `<sid>adm`). The corresponding UNIX environment variables are as follows:

- `LD_LIBRARY_PATH`: Solaris, Sinix, OSF/1, Rleiant UNIX, Digital UNIX
- `SHLIB_PATH`: HP-UX
- `LIBPATH`: AIX

Maintaining the System PSE

Use

The system PSE maintenance is available as of Release 4.5B.

Use the PSE maintenance to maintain and monitor the system PSEs.

You can:

- Generate a new system PSE and distribute it to the application servers.
- Import a local PSE and distribute it as the system PSE to all application servers.
- Change the PIN (Personal Identification Number) that protects the system PSE.
- Create credentials [Page 9] for the system PSE.

Procedure

1. To access the PSE maintenance, use transaction PSEMAINT.

   The *PSE Management* screen shows the status of all of the application servers' PSEs.

   The following statuses are possible:

   – **RFC**

     The status of the RFC connection to the application server can be one of the following:

     - `ok`
     - `contained errors`
     - `in waiting`

   – **PSE Status**

     The PSE and SSF status can be:

     - `PSE & SSF OK`

       This status indicates that a system PSE for the application server has been installed and accessible.

     - `SSF error`

       This status indicates that an external security product has been installed and that a system PSE exists on the application server; however, the system PSE cannot be accessed. The most common cause of this error is that no credentials exist on the application server. To correct this error, use the function *Create credentials*.

     - `No security product installed`

       This indicates that the SAPSECULIB has not been installed.

     - `Local PSE not installed`

This indicates that no system PSE exists on the application server. To correct this error, use either *Create PSE* or *Import PSE*.

- `PSE error`

  This error indicates that the version of the PSE on the application server does not coincide with the version that is stored in the database. To correct this error, use *Create PSE* or *Import PSE*.

2. Place the cursor on the application server where you want to execute the given function and choose the corresponding menu option.

   The following table shows the functions you can perform:

PSE Maintenance Functions

Function	Follow-on menu path	What you should know
Generate a new system PSE and distribute it to all of the application servers.	→ *PSE* → *Generate*	This function creates a new PSE on the chosen application server, imports it into the database, and distributes it to all of the remaining application servers. PSEs that already exist are overwritten.
Import a local PSE and distribute it as the system PSE to all application servers.	→ *PSE* → *Import*	This function imports a local PSE into the database and distributes it to all of the application servers. PSEs that already exist are overwritten.
Change the PIN that protects the system PSE.	→ *PSE* → *Change PIN*	The default system PSE is not protected with a PIN. We recommend you assign a PIN to protect the PSE.
Create credentials for the system PSE	→ *PSE* → *Create credentials*	The application server needs credentials to access its system PSE. Although credentials normally exist for an application server, occasionally you may have to create new ones, for example, for a newly configured application server that has not yet accessed its system PSE.
Function	**Use Push-Button**	**What you should know**
Change the list of certificates to use for verification	✏ *Certificate list*	With this function you can maintain a list of public-key certificates that can be used by the system to verify other users' or system components' digital signatures.

Export a version of the system PSE that can be used by others to verify the system's digital signatures.	*Verification PSE*	This PSE contains only the public information from the system PSE (for example, the system's public-key certificate and the system's public-key). This information can be distributed to others to be used to verify the system's digital signatures.
Issue a certificate request on the SAP CA.	*Certificate request*	With this function, a public-key pair and a public-key certificate are generated. The public-key certificate is then sent to the SAP CA to be signed.
Insert the certificate request response received from the SAP CA.	*Certificate request*	With this function, you import the returned response (signed public-key certificate) into the system.

Defining Default SSF Information for Applications

Use

Use this procedure to define a default set of SSF information to use for applications instead of the system defaults.

The following table shows the system defaults.

SSF Information System Defaults

SSF Information	Default Value	Comment
Security product	The product whose library is contained in the SSF profile parameter `ssf/ssfapi_lib`.	The default for this parameter is the SAPSECULIB library `libssfso`. You can change the value of this parameter during the SSF configuration. See Installing/Configuring SSF: Application Server [Page 15].
Name of the security product	The name of the product as specified in the profile parameter `ssf/name`.	`ssf/name` corresponds to the product specified in `ssf/ssfapi_lib.`
SSF format	PKCS#7	PKCS#7 is currently the only format supported.
Private address book	`<instance directory>/sec/SAPSYS.pse`	This is the location of the application server's PSE provided with SAPSECULIB, where `<instance directory>` is defined in the profile parameter `DIR_INSTANCE`.

Prerequisites

The name and location of the product's library must be specified in the profile parameter `ssf<x>/ssfapi_lib`. The product's name must be defined in the profile parameter `ssf<x>/name`.

Procedure

In the procedure Maintaining Application-Specific Information [Page 28], create or change the **`Default`** entry in the *Application* dropdown list.

Result

The system uses the values that you specify in the **`Default`** entry as defaults for the SSF information, instead of the system defaults.

Maintaining Application-Specific Information

Use

Use this procedure to specify application-specific information, for example, if you use more than one security product for using the SSF functions, or if you use the same product but different SSF profiles [Page 9] or private address books [Page 9] for different applications.

For example, if you use the ArchiveLink II HTTP content server 4.5 interface, which uses the SAPSECULIB as the security provider for signing archive requests, and you use an external security product for creating digital signatures in a different application, then you need to specify SSF information for each application.

Prerequisites

The security products have been installed on each application server.

The SSF profile parameters (or environment variables) `ssf<x>/name` and `ssf<x>/ssfapi_lib` contain the names of the security products and the names and locations of the products' libraries.

Procedure

1. To access the SSF information for specific applications, call transaction SSFA.

   If only one entry exists, the system displays the entry. Otherwise, it displays a table containing all existing entries.

2. If an entry for the application already exists and you want to change it, select it and choose *Goto* → *Detail*. Otherwise, to create an entry:

   a. Choose *Edit* → *New entries*.

   b. Select the application name from the dropdown list and press *Return*. (Choose or create the entry **`Default`** to create a default entry for applications. For more information, see Defining Default SSF Information for Applications [Page 27].)

      The maintenance screen for the entry appears.

3. Enter the following information in the corresponding fields:

   SSF Application-Specific Maintenance Fields

Field	Value	Comment

Security product	Name of the security product	The name of the product must match the name specified in one of the profile parameters `ssf<x>/name`. The system then uses the library specified in the corresponding profile parameter `ssf<x>/ssfapi_lib` for the application.
SSF format	PKCS#7	This is currently the only format supported.
Private address book	The name and location of the private address book	The private address book contains the public keys of the users and components. The name and location of the private address book is determined by the security product you use.
Profile name	The name and location of the SSF profile	The SSF profile contains the complete security information for the users and components (for example, the private keys). The name and location of the SSF profile is determined by the security product you use.

➡

The information you can maintain for an application is defined by the application. Therefore, only those fields that are necessary for the application appear.

➡

If you leave a field blank, the application uses the default value. The default value is determined either in the default entry, or, if no default entry has been defined, then the system defaults are used. (For more information, see Defining Default SSF Information for Applications [Page 27].) The currently defined defaults are displayed in the lower section of the maintenance screen.

4. Select *with certificate* if:
    a. Users' or components' certificates should be included with their digital signatures, or
    b. Certificates are to be used to verify digital signatures.
5. Select *only dig. signature* if the data that is signed is not to be included with the digital signatures.
6. Save the data.

Testing the SSF Installation

Use

After installing SSF, you can use reports SSF01 and SSF02 to make sure that the security product has been installed correctly. Use SSF01 to test the frontend installation and SSF02 to test the installation on the application server.

Procedure

1. Call transaction SE38.
2. To test the SSF installation on the front ends, enter **SSF01** in the *Program* field; to test SSF on the application server, enter **SSF02**.
3. Choose *Program* → *Execute* → *Direct*.

   The system displays a selection field of possible functions that you can test.
4. Select the *Version* function.
5. If you want to test the version for a specific security product, enter the product's name (as specified in the profile parameter `ssf<x>/name`) in the field *Security product* in the *Further options* group.

   ➡

   The other options and fields are not relevant for this test.
6. Choose *Program* → *Execute*.

   The system displays the results of the test. If the test was successful, the system displays the return code `SSF_API_OK` and version information about the security product. On the front end, it also displays the version of the SSF RFC server program. If the system encountered an error, it displays the error information.

   ➡

   If you use the SAPSECULIB as the security provider, then the test on the front end (report SSF01) will produce an error code. (SAPSECULIB is only installed on the application server.)

   The following return codes are possible:

SSF Test Report Error Codes

Error Code	Affects*	Definition	What to Do
SSF_API_OK	FE / AS	No error occurred.	

SSF_RFC_ERROR	FE	RFC destination is not correctly defined or the system cannot start the SSF RFC server program ssfrfc.exe.	Use transaction SM59. Make sure the RFC destination SAP_SSFATGUI is defined as a TCP/IP destination. Test the connection in SM59. Make sure that the file ssfrfc.exe exists in the correct location. (In a standard installation, it should be located in the SAP GUI directory.)
SSF_API_NO_SECTK	FE / AS	The security product is not installed correctly.	Make sure that SSF_LIBRARY_PATH (or ssf/ssfapi_lib) is set correctly. Set the trace level to 1 and check the contents of the trace file dev_ssf.

*FE=Front End; AS=Application Server

SSF Parameters

The following parameters define the SSF configuration on the front ends and on the application servers:

SSF Parameters

Parameter	Description
SSF_LIBRARY_PATH [Page 34]	Path and file name of the SSF library
SSF_MD_ALG [Page 35]	Message Digest Algorithm
SSF_SYMENCR_ALG [Page 36]	Symmetric Encryption Algorithm
SSF_TRACE_LEVEL [Page 37]	Trace level for recording SSF activities
SSF_NAME [Page 38] `(as of Release 4.5B)`	Name of the security product (application server only)

For complete descriptions and default values, see the documentation for each of the individual parameters.

Defining the SSF Parameters on the Front End Computers

On the front ends, you can specify the SSF parameters either in environment variables (as of Release 4.5A) or in the SSF initialization file `ssfrfc.ini`. For more information about using `ssfrfc.ini`, see The SSF Initialization File [Page 39].

Entries in the SSF initialization file override those in environment variables. If no entries exist in either environment variables or in the `ssfrfc.ini`, the system uses the default values.

Defining the SSF Parameters on the Application Servers

On the application servers, you can use either environment variables (as of Release 4.5A) or profile parameters to specify the SSF information. Profile parameters values override those in environment variables. If you do not use either option, then the system uses the default values.

Defining SSF Parameters when Using Several Security Products (Application Servers Only)

As of Release 4.5B, you can use up to three different security products for different applications. To differentiate between the SSF parameters for each application, there are three sets of SSF parameters that you can define on the application servers. The syntax for these parameter sets are as follows:

- **Environment variables:**
  - Product 1: `SSF_...`
  - Product 2: `SSF2_...`
  - Product 3: `SSF3_...`
- **Profile parameters:**
  - Product 1: `ssf/...`

- Product 2: `ssf2/...`
- Product 3: `ssf3/...`

Note however, that you can only install multiple security products on the application servers. You can only configure a single product on the front ends.

The SAP ArchiveLink II HTTP content server 4.5 interface uses SAPSECULIB as the security provider to create digital signatures. In Quality Management, an external security product `<product>` is used for signing inspection lots. This product also uses a different hash algorithm for creating digital signatures than SAPSECULIB. The following application server profile parameter definitions specify different values for each of these applications.

Example of Application Server Profile Parameters when Using Different Products

Application: SAP ArchiveLink II HTTP content server 4.5 interface

Product: SAPSECULIB

Parameter	Value
`ssf/ssfapi_lib`	Name and location of the SAPSECULIB library `libssfso`
`ssf/name`	SAPSECULIB
`ssf/ssf_md_alg`	MD5

Application: Quality Management

Product: `<product>`

Parameter	Value
`ssf2/ssfapi_lib`	Name and location of the product's library
`ssf2/name`	`<product>`
`ssf2/ssf_md_alg`	SHA1

In addition, use transaction SSFA to specify the `<product>` for the application **`quality certificate handling`**. For more information, see Maintaining Application-Specific Information [Page 28].

We describe the individual parameters in the topics that follow.

SSF_LIBRARY_PATH

Environment Variable or Initialization File Entry:

Product 1: `SSF_LIBRARY_PATH`
Product 2: `SSF2_LIBRARY_PATH`
Product 3: `SSF3_LIBRARY_PATH`

Application Server Profile Parameter:

Product 1: `ssf/ssfapi_lib`
Product 2: `ssf2/ssfapi_lib`
Product 3: `ssf3/ssfapi_lib`

Short Description: Path and file name of the SSF library

Valid Releases: All

Description: This parameter contains the complete path and file name of an external function library for the Secure Store & Forward functions. The name and the location of the library are determined by the security product you use.

Default: `libssfso.<ext>` (`<ext>` is the appropriate platform-specific extension. For example, under Windows NT, the extension is `dll`. Use `.XXX` to let the system automatically determine the correct extension.)

`libssfso.<ext>` is the SAPSECULIB library, which is the default security provider.

As the default on the front ends, the system searches for `libssfso.<ext>` in the directory where the executable program `ssfrfc.exe` is located. In a standard installation, this is the SAP GUI directory (for example, under Windows NT, the SAP GUI directory is `C:\Program Files\sappc\sapgui`).

On the application server, the default value of `ssf/ssfapi_lib` is empty, which means that the system searches for the SAPSECULIB library `libssfso.<ext>` in the directory specified by the profile parameter `DIR_LIBRARY`.

Valid entries, formats: Character string up to 255 characters

➡

You can only define the parameters for additional products on the application servers.

SSF_MD_ALG

Environment Variable or Initialization File Entry:	Product 1:	`SSF_MD_ALG`
	Product 2:	`SSF2_MD_ALG`
	Product 3:	`SSF3_MD_ALG`
Application Server Profile Parameter:	Product 1:	`ssf/ssf_md_alg`
	Product 2:	`ssf2/ssf_md_alg`
	Product 3:	`ssf3/ssf_md_alg`

Short Description: Message Digest Algorithm for SSF

Valid Releases: All

Description: This parameter contains the message digest algorithm to use with the SSF functions. The message digest algorithm is used, for example, when creating digital signatures.

You must specify an algorithm that is supported by the security product that you use.

Default: MD5

Valid entries, formats: MD2, MD4, MD5, SHA1, RIPEMD160

For other supported algorithms, see your security product's documentation.

You can only define the parameters for additional products on the application servers.

SSF_SYMENCR_ALG

Environment Variable or Initialization File Entry:	Product 1:	`SSF_SYMENCR_ALG`
	Product 2:	`SSF2_SYMENCR_ALG`
	Product 3:	`SSF3_SYMENCR_ALG`
Application Server Profile Parameter:	Product 1:	`ssf/ssf_symencr_alg`
	Product 2:	`ssf2/ssf_symencr_alg`
	Product 3:	`ssf3/ssf_symencr_alg`

Short Description: Symmetric Encryption Algorithm for SSF

Valid Releases: All

Description: This parameter contains the symmetric encryption algorithm to use with the SSF functions. The symmetric encryption algorithm is used, for example, when creating digital envelopes.

Note that you must specify an algorithm that is supported by the security product that you use.

Default: DES-CBC

Valid entries, formats: DES-CBC, TRIPLE-DES, IDEA

For other supported algorithms, see your security product’s documentation.

⇨

You can only define the parameters for additional products on the application servers.

SSF_TRACE_LEVEL

Environment Variable or Initialization File Entry:

Product 1: `SSF_TRACE_LEVEL`
Product 2: `SSF2_TRACE_LEVEL`
Product 3: `SSF3_TRACE_LEVEL`

Application Server Profile Parameter: Not defined as an application server profile parameter

Short Description: Trace level for recording SSF activities

Valid Releases: All

Description: Depending on the trace level, the system records information about SSF function calls in the file `dev_ssf`.

Default: 0 (lowest trace level)

Valid entries, formats: 0 : The system records:

- The starting of the SSF RFC server
- The loading of the SSF function library
- The installation of the RFC-enabled SSF functions

1 : The system also records:

- The name and the return codes of the SSF functions that are called

2 : The system also records:

- Information about the signer and receiver when SSF functions are called

3 : The system also records:

- All input and output data when SSF functions are called

You can only define the parameters for additional products on the application servers.

In addition, `SSF_TRACE_LEVEL` is not defined as a profile parameter on the application server. You can only specify its value with the environment variable.

SSF_NAME

Environment Variable:	Product 1:	`SSF_NAME`
	Product 2:	`SSF2_NAME`
	Product 3:	`SSF3_NAME`
Application Server Profile Parameter:	Product 1:	`ssf/name`
	Product 2:	`ssf2/name`
	Product 3:	`ssf3/name`
Short Description:	Name of the security product	
Valid Releases:	As of Release 4.5B	
Description:	This parameter contains the name of the security product.	
Default:	Product 1:	SSF
	Product 2:	SSF2
	Product 3:	SSF3
Valid entries, formats:	Character string up to 255 characters	

You can only define the parameters for additional products on the application servers.

`SSF_NAME` is also only defined on the application server.

When an application server is started, the system automatically loads the SAPSECULIB and assigns the SAPSECULIB information to the next available `ssf<x>/...` parameter set. For example, if no other product has been specified in the SSF parameters, the parameter `ssf/name` is set to the value SAPSECULIB when the application server is started.

The SSF Initialization File

Definition

File that contains the SSF configuration on the frontend computers.

Use

You can use the SSF initialization file to configure the SSF parameters on the front ends.

As of Release 4.5, you can use either the SSF initialization file or environment variables.

Name and Location

The default name of the SSF initialization file is `ssfrfc.ini` and is located in the directory where the SSF RFC server program `ssfrfc.exe` is located. (In a standard installation, this is the SAP GUI directory.) You can specify a different name and location with the environment variable `SSF_INI_FN`.

Syntax

The format for entering each parameter in the file is a line entry with the following syntax:

```
<SSF parameter>=<parameter value>
```

Note the following:

- Blank lines and lines that begin with an asterisk (*) are considered comments.
- If the SSF process of analyzing the file discovers an error, it terminates. The line where the error was found is recorded in the SSF trace file `dev_ssf`.

The following shows an example of the `ssfrfc.ini` file in Release 4.5.

ssfrfc.ini

```
****************************************************
* Secure Store & Forward (SSF) Initialization File *
****************************************************
SSF_LIBRARY_PATH=c:\Program Files\SAPpc\SAPgui\libssf.dll
SSF_MD_ALG=MD5
SSF_SYMENCR_ALG=DES-CBC
SSF_TRACE_LEVEL=0
```

Information Specific to Release 4.0/4.5

Certain maintenance tasks and SSF information have changed between the Releases 4.0, 4.5, and 4.6. The user SSF information maintenance has moved to user address maintenance and the use and syntax of the SSF initialization file has changed. For more information, see:

- Maintaining User SSF Information: Release 4.0/4.5 [Page 41]
- The SSF Initialization File in Release 4.0 [Page 42]

Maintaining User SSF Information: Release 4.0/4.5

Prerequisites

The security product has been installed and SSF has been configured on the application server (see Installing/Configuring SSF: Application Server [Page 15]).

The location of the SSF RFC server program `ssfrfc` also needs to be defined as the RFC destination `SAP_SSFATGUI` in Transaction SM59.

Procedure

1. In Customizing for *System Administration*, choose *Management of External Security Systems* → *Secure Store and Forward (SSF)* → *Maintain SSF-Information for the User* (transaction O07C).
2. Modify an existing entry by selecting the entry and choosing *Goto* → *Details*, or create a new entry by choosing *New entries*.

   The *Change View "Digital signature: SSF information about user": Details* screen appears.
3. Enter the corresponding values:

Maintenance of User SSF Information

Field	Description	Comment
Destination	The RFC destination (logical destination) where the SSF RFC server program has been defined.	SAP_SSFATGUI See the input help (F4).
S/R name	SSF Signer/Recipient name	The syntax of this name is determined by the security product you use.
Namespace (Release 4.0 only)	Namespace where the user's information is valid.	For example, the user's information may be stored on a smart card, in a local directory, or an X.500 directory. See the input help (F4) for possible entries.
Prof.name	Profile name	Specifies the profile where the user's security information is stored. The value of this parameter is determined by the security product you use.

4. Save the data.

The SSF Initialization File in Release 4.0

This topic explains the few differences that you need to consider when using the SSF initialization file in Release 4.0. For general information about using the file, see the topic The SSF Initialization File [Page 39].

No Environment Variable Support in Release 4.0

In Release 4.0, you have to use the SSF initialization file to configure the SSF parameters on the front ends.

Syntax for ssfrfc.ini in Release 4.0

In Release 4.0, the format for entering each parameter in the file is a line entry with the following syntax:

```
<SSF parameter description> = <parameter value>
```

Note the following:

- The SSF parameter descriptions in Release 4.0 are case-sensitive. You also have to include exactly one space before and after the equal sign (=). The descriptions are as follows:
  - `SSF Library Path`
  - `SSF Hash Algorithm`
  - `SSF Symmetric Encryption Algorithm`
  - `SSF Trace Level`
- The last line of the file must contain an end return.

The following shows an example of the `ssfrfc.ini` file in Release 4.0.

ssfrfc.ini

```
*****************************************************
* Secure Store & Forward (SSF) Initialization File *
*****************************************************
SSF Library Path = c:\Program Files\SAPpc\SAPgui\libssf.dll
SSF Hash Algorithm = MD5
SSF Symmetric Encryption Algorithm = DES-CBC
SSF Trace Level = 0
<Return>
```