

# **Digital Signatures in SAP Applications**



**SAP Web Application Server 640**



## Copyright

© Copyright 2003 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft<sup>®</sup>, WINDOWS<sup>®</sup>, NT<sup>®</sup>, EXCEL<sup>®</sup>, Word<sup>®</sup>, PowerPoint<sup>®</sup> and SQL Server<sup>®</sup> are registered trademarks of Microsoft Corporation.

IBM<sup>®</sup>, DB2<sup>®</sup>, DB2 Universal Database, OS/2<sup>®</sup>, Parallel Sysplex<sup>®</sup>, MVS/ESA, AIX<sup>®</sup>, S/390<sup>®</sup>, AS/400<sup>®</sup>, OS/390<sup>®</sup>, OS/400<sup>®</sup>, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere<sup>®</sup>, Netfinity<sup>®</sup>, Tivoli<sup>®</sup>, Informix and Informix<sup>®</sup> Dynamic Server<sup>™</sup> are trademarks of IBM Corporation in USA and/or other countries.

ORACLE<sup>®</sup> is a registered trademark of ORACLE Corporation.

UNIX<sup>®</sup>, X/Open<sup>®</sup>, OSF/1<sup>®</sup>, and Motif<sup>®</sup> are registered trademarks of the Open Group.

Citrix<sup>®</sup>, the Citrix logo, ICA<sup>®</sup>, Program Neighborhood<sup>®</sup>, MetaFrame<sup>®</sup>, WinFrame<sup>®</sup>, VideoFrame<sup>®</sup>, MultiWin<sup>®</sup> and other Citrix product names referenced herein are trademarks of Citrix Systems, Inc.

HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C<sup>®</sup>, World Wide Web Consortium, Massachusetts Institute of Technology.

JAVA<sup>®</sup> is a registered trademark of Sun Microsystems, Inc.

JAVASCRIPT<sup>®</sup> is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MarketSet and Enterprise Buyer are jointly owned trademarks of SAP AG and Commerce One.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves information purposes only. National product specifications may vary.

## Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

## Typographic Conventions

Type Style	Description
<i>Example text</i>	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.
<b>Example text</b>	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.
<b>Example text</b>	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.

Copyright .....	2
Icons .....	3
Typographic Conventions .....	3
Digital Signatures .....	6
1. Technical Prerequisites.....	8
1.1. Signature Types .....	8
1.1.1 User Signature with Own Key Pair .....	8
1.1.2 User Signature with System Key Pair (System Signature).....	8
1.1.3 Server Signature .....	9
1.1.4 Server Signature Linked to a Natural Person .....	9
1.2. Signature Format.....	9
1.2.1 PKCS#7 .....	10
1.2.2 XML .....	10
1.2.3 Adobe .....	10
1.2.4 S/MIME .....	11
1.3. SSF.....	11
1.3.1 SSF for the ABAP Stack.....	11
1.3.2 SSF for Java .....	12
1.3.3 Interfaces .....	12
1.4. Display Components .....	13
1.4.1 SAP Signature Control .....	13
1.4.2 External Signature Display Component.....	14
1.4.3 Adobe .....	14
1.5. Key Management .....	14
1.6 Secure Signature-Creation Devices .....	15
1.7 Signature Checking.....	15
2. Implementation .....	16
2.1 Determining the Signature Type .....	16
2.2 Integration into the Application.....	17
2.2.1. SAP Business Workflow .....	18
2.3 Storing the Digital Signature .....	18
3. Conditions .....	21
3.1 SmartCard for Authentifizierung und digitale Signatur.....	21
3.2 Signatures Act.....	21
3.2.1 Electronic Signatures.....	21
3.2.2 Signature-Application Components.....	22

---

3.2.3 Manufacturer's Declaration.....	23
3.2.4 Validity Period of Qualified Certificates .....	23
3.3 Europe-Wide Implementation of the Signature Guidelines .....	24
3.4 ISIS-MTT .....	24

## Digital Signatures

Digital signatures provide cross-enterprise transactions with electronic characteristics that can be used to prove both the authorship and the original content of the data over a long period of time.

In many countries, there are already laws that regulate the prerequisites for a legal equivalence between digital and handwritten signatures.

- The European Union created common prerequisites for electronic signatures for its member states in 1999 with the Signature Guidelines. The aims of the guidelines were to be incorporated into national law in all EU countries from a technical point-of-view by 19 July 2001. The implementation of the guidelines in Germany was completed when the new Signatures Act came into force on 22 May 2001. Since the guidelines also deal with the equivalence of electronic signatures with traditional signatures, other laws must also be modified to fully implement the guidelines. With the exception of a few adjustments in administrative law, these have also been completed to a large extent.
- In the USA, a similar law, "Electronic Signatures in Global and National Commerce Act", has been in force since October 2000.
- The American Food and Drug Administration (FDA) of the American Department of Health and Human Services has specified in the "21 Code of Federal Regulations (CFR) Part 11" that an electronic signature must be added to documents in the case of purely electronic-based document management. Companies that produce in the USA, and European companies that deliver to the USA must adhere to the code. For the European Union, there are implementations such as Annex 11 of the EC Good Manufacturing Practice (GMP) guidelines, the German Ordinance on Pharmaceutical Businesses (Pharma-Betriebsverordnung), and the German Drug Law (Arzneimittelgesetz).

The advantages of digital signatures are:

- It is possible to uniquely identify the sender of a message (authenticity)
- It is possible to determine whether the original content of the message is unchanged (integrity)
- It is possible to check whether the user's digital signature is still valid
- Digital signatures cannot be reused, since they are part of the document
- The sender cannot later dispute his or her agreement

The signature can be created both by the server and by the user. For example, digital signatures by servers are used, in accordance with a regulation of the American Food and Drug Administration (FDA), in the SAP solution for the process industries to secure recipes. Electronic signatures by the server can also be used for the mass processing of documents, such as bills (for Europe, with a connection to a natural person, such as a procurist). SAP offers this solution with the Exchange Infrastructure. Digital signatures by users are used, for example, in the public sector for communication between constituents and authorities. Digital signatures are also used for mobile business processes: in this

way, doctors can confirm the receipt of drug samples from representatives of pharmaceutical companies using a digital signature on a PDA device.

# 1. Technical Prerequisites

## 1.1. Signature Types

You must decide, using the process to which you want to assign a digital signature, which type of signature you want to implement. There are essentially four types of signature.

### 1.1.1 User Signature with Own Key Pair

With this signature, the user has his or her own key pair and a digital certificate issued by a trusted Certification Authority (CA) with the public key that it has certified. The private key can be stored, for example, on the hard disk, or on a smart card.

If the digital signature is to meet the requirements for a qualified electronic signature as defined in the German Signatures Act, the certificate must be issued by a certification service provider approved by the RegTP (Regulierungsbehörde für das Telekommunikations- und Postwesen [Regulatory Authority for Telecommunications and Posts]), and a secure signature-creation device (SmartCard/SmartToken) must be used (for a more detailed description, see section 1.6, Secure Signature-Creation Devices).

The typical signature process for user signatures must display the document to be signed for the user at the front end, and the user starts the signature process after checking the document (visual inspection), for example, using a sign button on the display dialog box. The user is actively and immediately involved in the signature process. This signature process represents the highest level of security and is independent of the security of the system itself.

This procedure can only be used if a public-key infrastructure exists.

### 1.1.2 User Signature with System Key Pair (System Signature)

If the signature process uses this signature type, the user does not have his or her own key pair for the signing of the document. In this case, the system's key pair is used. The document is displayed directly to the user using an appropriate display component (viewer), and the user actively starts the signature process. Instead of the user's key pair, the server queries and checks the user's password. Only after this check has been successfully performed is the document signed with the server's key.

This procedure meets the requirements of the FDA, but not of the German Signatures Act. An additional restriction is that this procedure can only be implemented in a local system. It only works to a limited extent, for example, for a portal solution, since it is not the user's portal password that is checked, but rather his or her password in the back end system. The user must therefore be clear that he or she must enter the password for the back end system. You must now clarify whether this is reasonable for your users.

With this scenario, the security of the signature process is directly dependent on the system security. If, for example, a user has too many authorizations on the system and can write reports that generate a signature, he or she could sign any documents.

The system signature (in the SAP system) is implemented in such a way that the user name is attached to the document and forms part of the signed document. This means that the process that checks the signature understands the user name extension. This procedure is proprietary.

This procedure does not require an existing public-key infrastructure, since only the server requires a key pair.

### 1.1.3 Server Signature

SAP refers to a server signature, if the server signs documents with its own key pair, independently of the user. This means that no display component is required for the user. Typical implementation areas for this signature are mass signatures, or signature processes that run as background processes. The document to be signed is sent to the server and it signs the document with its own key pair without prior interaction with the user. An application example already in use is the signing of IDocs.

This procedure does not meet the requirements of the Signatures Act, since the signer must be a natural person. The main use of this procedure is to achieve a higher level of security for background or mass processes.

### 1.1.4 Server Signature Linked to a Natural Person

For this procedure, a natural person makes his or her certificate available for a server signature. A procurist must usually request a certificate from an approved certification service provider and make this available for automatic signatures. It is, of course, possible to ensure within the company, for example, with a contract, that the procurist is not personally liable for all of these signatures. For performance reasons, it is, naturally, possible to use multiple SmartCards. The advantage of this type of signature is that, as long as a number of conditions are met (such as a secure system environment), it meets the requirements of the German Signatures Act.

## 1.2. Signature Format

SAP currently supports four different signature formats. The signature format essentially depends on the application.

Product/Signature Format	PKCS#7	XML	S/MIME	PDF (Adobe)
SAP Signature Control	+	-	-	-
SSF Partner Product	+	-	-	-
SAP Java SSF Library	+	+	+	-
Adobe Acrobat	-	-	-	+
System Signature	+	-	-	-

### 1.2.1 PKCS#7

PKCS#7 stands for Public-Key Cryptography Standard # 7. The PKCS standards are specifications that were developed by RSA Security for secure information exchange using the Internet. PKCS#7 is currently a format established in the market. It describes a wrapper format, meaning that the output format does not correspond to the input format. If, for example, a PKCS#7 signature is attached to a PDF document, the document is then in PKCS#7 format. This format is a binary format, which in turn means that direct display is not very good, and is not suitable for a user. A special tool (viewer) must therefore be used for display. Unfortunately, usable PKCS#7 viewers are not currently available on the market. This format is very suitable for purely automatic processing (without a display component for the user).

SAP offers the PKCS#7 signature on the ABAP platform (as of SAP Basis 4.0B) and on the Java platform (as of SAP Web AS 6.30).

### 1.2.2 XML

Signatures in XML format are only now becoming acceptable in the market.

With an XML signature, the document to be signed, and the signed hash value are stored in an XML document. The document can contain the XML signature, the XML signature can enclose the document, or the XML signature can be stored separately from the XML document. This means that any parts of an XML document can be signed.

A significant advantage of the XML signature is the visualization component. XML is made up of readable characters and can be extended using XSL (Extensible Style Language). If the stylesheet (the description of the presentation of the XML data) is also signed, the formatted document can be displayed to the user in any browser. The fact that the stylesheet is also signed ensures that the display of the data corresponds to the signed XML data.

SAP currently offers this format only on the Java platform.

### 1.2.3 Adobe

Digital signatures can also be created and verified as of Adobe Reader 5.1. Users can use the Adobe Acrobat Reader to display and print files in PDF format, whose content and appearance correspond to the paper version, on a cross-platform basis. The prerequisites for this are the use of the new Adobe Document Server for Reader Extensions and the replacement of SmartForms by Adobe Acrobat (this is still in development). SSF is not used.

Adobe Document Server for Reader Extensions assign additional usage rights to electronic forms that were created with Adobe Acrobat 5.0 and Adobe Form Designer 5.0. Digital signatures are a type of usage right.

Adobe Acrobat Reader 5.1 can be downloaded free-of-charge from Adobe's Web site, and processes the usage rights that were embedded in PDF forms by the Adobe Document Server for Reader Extensions.

## 1.2.4 S/MIME

S/MIME is the standard for signing e-mails. S/MIME is based on the PKCS#7 data format and on the X.509v3 format for certificates. SAP offers this format only on the Java platform.

## 1.3. SSF

Secure Store and Forward (SSF) mechanisms allow the creation and checking of digital signatures. Documents can also be encrypted and decrypted using digital envelopes.

A prerequisite for successfully using SSF mechanisms is that a public-key infrastructure (PKI) has been set up. A PKI ensures that digital signatures, certificates, and certification instances can be checked. For a more detailed description, see section 1.7, Signature Checking. Although SAP systems do not contain any PKI functions, they support the PKI provided by various security products.

SSF is provided for SAP Web Application Server 6.30 for the ABAP stack and for the J2EE. This is done using two different libraries, about which details are provided below.

Format	Product
PKCS#7	SAP Java Cryptographic Toolkit (Java) SSF partner product (ABAP)
S/MIME Version 2	IAIK S/MIME (Java)
XML	SAP XML Toolkit (Java)

SAP Java Cryptographic Toolkit, IAIK S/MIME, and SAP XML Toolkit are delivered in SAP J2EE Server 6.30. They contain only signature functions and no functions for encryption and decryption. The toolkits must be downloaded from the SAP Service Marketplace for encryption and decryption.

### 1.3.1 SSF for the ABAP Stack

The SSF Library for the ABAP Stack is used in applications that are written in ABAP. It supports the functions for creating and verifying digital signatures (PKCS#7), and functions for encrypting and decrypting documents.

SSF requires an external security product to provide these functions. The SAP Security Library (SAPSECULIB) is delivered with the SAP system as the default product. However, the SAP Security Library only supports digital signatures without cryptographic hardware (SmartCards, SmartTokens, Cryptoboards). Instead of the SAPSECULIB, customers can also use the SAPCRYPTOLIB, which can be downloaded from the SAP Service Marketplace. The SAPSECULIB supports the DSA (Digital Signature Algorithm) algorithm, and the SAPCRYPTOLIB supports both the DSA and the RSA algorithms. The algorithm that you must use in your signature process depends on the CA that issues the certificate. Most CAs use the RSA algorithm. Note that country-specific export guidelines apply in the case of the SAPCRYPTOLIB. For more information, see SAP Note 397175.

For support for encrypting and decrypting documents, and for generating digital signatures using cryptographic hardware, an external security products from our partners is required. These security products use SAP's SSF interface and are certified for this by SAP. For a list of the certified products, see the SAP Service Marketplace under <http://service.sap.com/securitypartners>, and then choose the link "Partner for Secure Store and Forward, digital signatures" (SSF).

The SSF Library for the ABAP stack is available as of SAP Basis 4.0.

## 1.3.2 SSF for Java

The Java SSF Library supports all Java applications. It is delivered with SAP J2EE Server 6.30 and is based on the IAIK Toolkit of Graz Technical University. This SSF Library also supports the generation and verification of digital signatures. To encrypt and decrypt documents, the IAIK Toolkit, which can be downloaded from the SAP Service Marketplace, must be installed. Therefore, no external security product is required from our partners, and there is no SAP certification program for the Java SSF Library.

PKCS#12 and the Java Keystore are supported for key storage, meaning that it is currently only possible to generate digital signatures without cryptographic hardware. Unlike the library for the ABAP stack, the Java library also supports XML and S/MIME signatures in addition to PKCS#7 signatures.

## 1.3.3 Interfaces

### 1.3.3.1 Interfaces for the ABAP Stack

SSF provides the following ABAP function modules from the SSFG function group:

- SSF Sign                      Creating digital signatures
- SSF Verify                    Checking digital signatures
- SSF Envelope                Encrypting documents
- SSF Develop                 Decrypting documents

For a detailed description, see [www.service.sap.com/security](http://www.service.sap.com/security) → Security in Detail → Digital Signatures → Secure Store and Forward (SSF) Programmers' Guide.

### 1.3.3.2 Interfaces for Java

The following interfaces and classes are supported:

- Data format

There is a central interface `IssfData` that contains all basic methods such as `sign`, `verify`, `encrypt`, and `decrypt`. This interface is implemented for the different classes (`SsfDataPKCS7`, `SsfDataSMIME`, `SsfDataXML`) that provide specific methods for each format.

- Key access

An SSF profile that contains the private key and the certificate's upward path is required for signing and decryption. An SSF Public Address Book (PAB) that contains a list of trusted root certificates is required for verification and encryption.

- XML-specific

Help classes for implementing the XML signature.

For a detailed description of the interfaces and classes, see the SAP Online Help under "Interfaces and Classes for Using Digital Signatures and Encryption".

### 1.3.3.3 SSF Profile

The "SSF profile" is a parameter that is used by SSF functions that require the private key. In the case of the digital signature, this would be the function SSF\_Sign for creating a digital signature. The SSF profile is the identifier of a file or a SmartCard that specifies the public-key information such as the private key and the certificate's upward path. The exact format of the profile depends on the security product in use.

For more information, see the SAP Help Portal <http://help.sap.com> → SAP NetWeaver → SAP Web Application Server → Security → Secure Store & Forward / Digital Signatures.

### 1.3.3.4 Private Address Book (PAB)

The "Private Address Book" is a parameter that is used by SSF functions that require a list of trusted certificates. For the digital signature, this would be the SSF\_Verify function for verifying the digital signature. The way in which the "Private Address Book" is stored in the system depends on the security product in use. With the Signature Control, Microsoft Keystore is used, and for the SAP J2EE Engine, the Keystore Service is used.

For more information, see the SAP Help Portal <http://help.sap.com> → SAP NetWeaver → SAP Web Application Server → Security → Secure Store & Forward / Digital Signatures.

## 1.4. Display Components

Display Component	SAPGUI for Windows	Business Server Pages	Web Dynpro
SAP Signature Control	+ (as of SAP GUI for Windows 6.40)	+ (as of SAP Web AS 6.20)	-
External Display Component	+	-	-
Adobe	-	-	+

### 1.4.1 SAP Signature Control

The signature control is used to display the document to be signed to the user at the front end, and to evaluate the signature. It can be called by the SAP GUI for Windows or using

Business Server Pages (BSP). The document is displayed in the signature control in text, HTML, or XML format, or as a PDF.

The SSFS function group is available for this purpose:

- SSFS\_Call\_Control: Starts the signature control
- SSFS\_Get\_Signature: Fetches the signature value from the control
- SSFS\_Server\_Verify: Verification of the digital signature by the application server

Strings and XStrings are used as data types.

The signature control is based on the Microsoft Crypto API. If MS-CAPI is used, it is also possible, for example, to include SmartCards using corresponding drivers, and no external security product is required. Alternatively, the signature control can load the SSF Library; an external SSF partner product is required to do this.

Every SAP Web Application Server 6.30 contains the report SSFSDEMO. This report shows how the functions SSFS\_Call\_Control and SSFS\_Server\_Verify are called.

## 1.4.2 External Signature Display Component

It is possible to use an external display component from our SSF partners. The external signature display component can be called by the function SSF\_Sign. If you transfer an exclamation mark as a parameter for the password field in the SignerList of the SSF\_Sign function, the external software queries the password. The data to be signed can then be displayed.

For a more detailed description, see [www.service.sap.com/security](http://www.service.sap.com/security) → Security in Detail → Digital Signatures → Secure Store and Forward (SSF) Programmer's Guide.

## 1.4.3 Adobe

Adobe Acrobat Reader 5.1 can be downloaded free-of-charge from Adobe's web site, and processes the usage rights that were embedded in PDF forms by the Adobe Document Server for Reader Extensions. It is planned to make it possible to use Adobe Acrobat Reader 5.1 as the display component in SAP applications.

## 1.5. Key Management

The Trust Manager is delivered as of SAP Web AS 6.10 for managing the public-key information at the server for the SAPSECULIB and the SAPCRYPTOLIB. The Trust Manager performs the Personal Security Environment (PSE) and certificate maintenance functions, such as generating key pairs, creating certificate requests that are to be signed by a Certification Authority (CA), and maintaining the list of trusted CAs that the server accepts.

For more information about the Trust Manager, see the SAP Help Portal <http://help.sap.com> → SAP NetWeaver → SAP Web Application Server → Security → Trust Manager.

The J2EE Engine supports Java's keystore management, the Java Keystore. The Keystore has key and certificate entries for the server. The Keytool, which has a command line interface, is available for administering the Java Keystore. For more information, see your Java documentation.

Alternatively, you can use a PKI administration tool from a third-party vendor, for example, for administering the users' public-key information.

## 1.6 Secure Signature-Creation Devices

The Signatures Act requires that secure signature-creation devices are used for storing signature keys and for creating signatures for qualified electronic signatures. These devices must satisfy the EAL 4 evaluation or the E 3 evaluation at "high". For a list of the accredited secure signature-creation devices, see [www.regtp.de](http://www.regtp.de) → Electronic Signature → Electronic Signature Products.

## 1.7 Signature Checking

A signed document contains the signature value and the digital certificate of the signer. During the check of the digital signature, cryptographic algorithms are used on the document and the signature value. If the check is successful, the document is unchanged, and was signed using the private key that belongs to the signer. An additional step of the check is to check whether the signer's certificate is valid, and whether the Certification Authority (CA) that issued the certificate is trusted. This is done using the Private Address Book (PAB). The next check is to ensure that the certificate has not been revoked by the CA, for example due to the loss of the SmartCard. The CAs issue a list of certificates that have been revoked or declared invalid (Certificate Revocation List - CRL). The CRLs can either be directly downloaded from the CAs during their valid time period, which must be configured at the customer site, or the Online Certificate Status Protocol (OCSP), which is based on the Hypertext Transfer Protocol (HTTP), can be used. This new standard is faster, simpler to use, and more reliable than CRLs. The application sends a request to a server application, the OCSP Responder, which maintains current status information and sends a response of valid, invalid, or unknown. OCSP services must usually be purchased by the customer. An exception to this is the used of qualified certificates. Most accredited trust centers offer this service free-of-charge for qualified certificates, since they cannot use CRLs due to the requirements of the German Signatures Act.

## 2. Implementation

### 2.1 Determining the Signature Type

You must decide which signature type you want to implement. The following types are available:

- User signature with own key pair
- User signature with the system's key pair
- Server signature
- Server signature linked to a natural person

A number of conditions restrict the selection of the signature type:

- The signature is to meet the requirements of the European Signature Guidelines.  
You can use only "user signature with own key pair" or "server signature linked to a natural person". The "server signature linked to a natural person" is used for automatic signing and verification processes, and "user signature with own key pair" is used for signature processes with display components. You should also differentiate between whether an advanced electronic signature or a qualified electronic signature, as defined by the German Signatures Act, is to be created (see section 3.2.1 Electronic Signatures). For an advanced electronic signature, it is sufficient to use software-based certificates. The qualified electronic signature can only be created with a secure signature-creation device (see section 1.7 Secure Signature-Creation Devices), such as a SmartCard or a SmartToken (corresponding Hardware Security Modules [HSM] are not yet available on the market).
- The signature is to meet the requirements of the FDA.  
The CFR Part 11 specifies that, in the case of purely electronic document management, a digital signature must be attached to the documents. In a closed system (such as an SAP R/3 system), a simple electronic signature is sufficient; in open systems, a qualified electronic signature is required.
- The signature is to be automatically created/generated (in the background).  
You can choose between "server signature" and "server signature linked to a natural person". You would usually only use the latter variant if a signature solution in accordance with the European Signature Guidelines is required. The "server signature" is sufficient in all other cases. Mass signing is a signature process that is performed by a technical user, meaning that the system's key pair can be used.  
An existing PKI is not required, since every SAP server has its own key pair, which is generated during installation.
- The signature process is implemented in an environment without a PKI.  
If no PKI is available, the signatures with person-linked key pairs are not possible.

## 2.2 Integration into the Application

You can implement your own extensions, such as the signature process, using Business Add-Ins (BADI), Customer Exits, or your own modifications. In general, the method using your own modifications should only be adopted, if no other possibilities are available. A modification always means that there are delays during a later upgrade of the SAP system. This is not the case, however, with BADIs and Customer Exits. SAP developers have provided special exits for customer to include customer extensions. Business Add-Ins are available for this purpose as of SAP Basis 4.6B, and Customer Exits are available in older releases. The available Business Add-Ins are usually described by the individual applications, but you can also search for them using transaction SE18 or transaction SPRO in the SAP system. If you do not find a suitable Business Add-In, you can investigate whether there are Customer Exits that meet your requirements. You can use transaction CMOD to search existing Customer Exits.

One of the most complex steps of integrating the signature process into the application is to find out where and how you insert your extensions. You must be clear about the application and the business process to which you want to assign the digital signature.

You should also, of course, clarify that the SAP application does not yet provide the function by default. For an overview of the applications that provide the digital signature function, see <http://service.sap.com/security> → Security in Detail → Digital Signatures → “Transactions and Processes Using Digital Signatures”. Future processes are also described in the “Technical Brief: Digital Signatures in Practice”. If you cannot find your process there, you can send a mail to [security@sap.com](mailto:security@sap.com) with a description of your signature scenario to make sure that no SAP standard solution exists or is planned.

You will usually build the signature extensions into an existing document flow, for example, in a defined interface after the creation of a document (order, material requests, and so on), redirecting the printer output, or as an intermediate step in IDoc processing.

Note that in signature scenarios that meet the requirements of the German Signatures Act, the document must always be digitally signed or verified in its original form. This determines where the integration should take place, since the signature should usually take place at the end of the document flow, and verification should usually take place at the start of the document flow.

Another aspect to consider is logging. For reasons of traceability, it would be useful if the application with the signature process creates a log containing the following information:

For the signature creation:

- When the signature was created
- Who created the signature
- What was signed

For signature checking:

- When the verification was performed
- What was verified (content of the signed data)
- Result of the verification

### 2.2.1. SAP Business Workflow

It is always possible to use the workflow for digital signatures if there is a flexible process flow with different users. For example, one user parks the data and another user releases the data using a digital signature. The automatic log is an additional advantage of the SAP Business Workflow. Other possible application areas of the workflow are multiple approval steps, different specific users for the digital signature, or hierarchy mappings, such as the manager of user X also having to sign, after all individual approval steps have already been signed.

An ideal solution would be to implement a generic decision workflow, with a button for creating a digital signature. This is not currently available in the standard system, and you must therefore implement it yourself.

The following section describes the implementation in broad terms:

Two workflows are required: one for the data input and one for the data release with a digital signature. The workflow for the data input consists of a method for creating the data and assigning users for the data input. The approval workflow consists of a method for data release. If there are multiple users that have to approve, the workflow defines the order and generates the digital signature using the SSF Library. In summary, the following implementation steps are required:

1. A method for data creation
2. A method for approval
3. A workflow for data creation
4. A workflow for approval

The methods are coded routines. It must be possible to call the dialog input using a URL. It can be implemented, for example, with a screen, iViews, BSP, a Web Dynpro, PHP, Java, HTML, or XML. The relevant workflows are created using the Workflow Builder.

For more information, see <http://help.sap.com> → SAP NetWeaver → SAP Web Application Server → Business Management → Trust Manager

## 2.3 Storing the Digital Signature

Assigning a digital signature to a process step means that the created signature has to be stored. The way in which the signature is stored depends on the application and the conditions of the project.

If the signature process is to meet the requirements of the German Signatures Act, the document must be stored in its original form on a medium that can only be written once, such as an optical archive. If the document is changed in other process steps, it must also be possible to access the document in its original form.

An additional point is long-term archiving. If you are legally obliged to archive the signed document for the long-term, the documents must be assigned a new qualified electronic signature before the expiration of the suitability of the algorithms, or before the time limit of five years. So that earlier qualified electronic signatures cannot be disputed based on possible forgery possibilities in the future, these must be included in the new signature, and therefore “conserved”. To avoid new qualified electronic signatures being added at a later date and back-dated, if the security value of the earlier digital signature may have

become so low that forgeries are possible, a qualified time stamp is required for this signature.

The new electronic signature and the qualified time stamp do not need to be generated individually for each electronically signed document. In fact, many documents can be re-signed electronically with one signature (this is based on section 17 of the Ordinance on Electronic Signatures, due to the plural uses of "data" and "earlier signatures").

Extract of the Ordinance on Electronic Signatures, section 17: Period and procedure for long-term data security:

Pursuant to Section 6 (1) sentence 2 of the Signatures Act, data with a qualified electronic signature shall be re-signed if they are required in signed form for a period longer than that for which the algorithms and related parameters used to create and verify them are considered to be suitable. In this case the data shall be furnished with a new qualified electronic signature prior to the time at which the suitability of the algorithms and related parameters ends. This signature shall be furnished with suitable new algorithms or related parameters, include earlier signatures and bear a qualified time stamp.

If there are no legal requirements for the signature process, the signature or the signed document could be stored in a database table, an archive, or a file.

SAP NetWeaver provides ArchiveLink, the Archive Development Kit (ADK), and the Knowledge Warehouse (KW) for archiving documents.

Storing data using a background process will only be supported by default in the Knowledge Warehouse as of release 8.0. In earlier KW releases, this is only possible on a project basis. Auditable storage of the data is supported. Since the required function of process-controlled archiving is not yet available in the standard system, archiving in the Knowledge Warehouse is not described in any further detail in this guide.

ArchiveLink is a document management interface to external storage and archiving systems and a service that allows SAP applications to store and search documents in an auditable process, using different methods.

The Archive Development Kit (ADK) is a subset of data archiving and is used for archiving database tables.

You must now decide in your project, whether the signed data must be stored in an auditable way. Both SAP Archive Link and the ADK can control external storage systems, such as an optical archive. If you do not want to archive in an auditable way, you can use the SAP Content Server (based on database technology) as the storage system. Both tools provide SAP transactions for accessing the archive.

The decision of whether to use the ADK or ArchiveLink depends on the signed data type. If the signed data is stored in a database table, you must use the ADK. Otherwise (PDFs, Microsoft Word files, and so on), you should use ArchiveLink.

Another aspect is the versioning of a document. It can be necessary in a project to attach multiple signatures to a document at different times. The versioning of archived documents is not supported by ArchiveLink or the ADK. An alternative would be to store multiple documents (different versions) under one Business Object.

Archiving Component	SAP ArchiveLink	SAP ADK	Knowledge Warehouse
Process-Controlled Archiving	+	+	-

			(only on a project basis)
Auditable Archiving	+	+	+
Data Type	Documents (PDF, Microsoft Word, and so on)	Database tables	All
Versioning	- (Possible using different documents)	- (Possible using different documents)	+

For more information, see [www.service.sap.com/archivelink](http://www.service.sap.com/archivelink).

## 3. Conditions

### 3.1 SmartCard for Authentication and Digital Signatures

Companies often want to use the same SmartCard for entering the company building, logging on to the computer, and for digital signatures. SAP supports this possibility in principle. Authentication for a SAP system can be performed using an X.509 certificate. The customer must simply ensure that the certificate on the SmartCard is sent to the browser using Microsoft CAPI (Internet Explorer) or PKCS#11 (Netscape / Mozilla). An external security product that supports MS-CAPI or PKCS#11 is required to do this.

If a logon to an SAP R/3 system with a SmartCard using the SAP GUI for Windows is desired, Secure Network Communication (SNC) must be installed between the front end and the back end. The external security product must therefore support SNC.

An external security product that supports SSF is required to create a digital signature without using the SAP signature control, or for the server signature with control of cryptographic hardware. Note that the SSF Library for Java cannot be used in this case, since this does not support any cryptographic hardware. As a consequence, only signatures in PKCS#7 format are supported if you are using cryptographic hardware.

If you are using the SAP signature control, you must use an external security product that supports MS-CAPI.

### 3.2 Signatures Act

The EU Signature Guidelines (19 January 2000) define the legal prerequisites for electronic signatures and certificate services. The German Signatures Act (Signaturgesetz, 22 May 2001) provides the conditions for electronic signatures in Germany (replacing the "Digital Signature Act" of 22 July 1997) and implements the European Parliament's guidelines and the directive on a community framework for electronic signatures of 18 November 1999. Details of the new Signatures Act as they apply are given in the Ordinance on Electronic Signatures (Signaturverordnung, 24 October 2001). The regulations in it are primarily implemented on the basis of the EU Signature Guidelines.

#### 3.2.1 Electronic Signatures

Term definitions in the German Signatures Act, section 2, paragraphs 1,2, and 3:

1. *"electronic signatures"*  
*Data in electronic form that are attached to other electronic data or logically linked to them and used for authentication;*
2. *"advanced electronic signatures"*  
*electronic signatures as in 1. above that*
  - a) *Are exclusively assigned to the owner of the signature code*
  - b) *Enable the owner of the signature code to be identified*

- c) *Are produced with means that the owner of the signature code can keep under his sole control and*
  - d) *Are so linked to the data to which they refer that any subsequent alteration of such data may be detected;*
3. *“qualified electronic signatures”*
- electronic signatures as in 2. above that*
- a) *Are based on a qualified certificate valid at the time of their creation and*
  - b) *Have been produced with a secure signature-creation device;*

An example of an “electronic signature” would be a digital signature with a user name and password; an example of an “advanced electronic signature” would be a digital signature with a software-based certificate; and an example of the “qualified electronic signature” would be a digital signature with a qualified certificate that is stored on a SmartCard.

### 3.2.2 Signature-Application Components

Term definitions in the German Signatures Act, section 2, paragraph 11:

*“signature-application components“ shall be software and hardware components designed to*

- a) *Assign data to the process of producing or testing qualified electronic signatures or*
- b) *Test qualified electronic signatures or check qualified certificates and display the results*

Excerpt from section 17 of the German Signatures Act, paragraph 2:

*The presentation of data to be signed requires signature-application components that will first clearly indicate the production of a qualified electronic signature and enable the data to which the signature refers to be identified. To check signed data, signature-application components are needed that will show*

1. *To which data the signature refers*
2. *Whether the signed data are unchanged*
3. *To which signature-code owner the signature is to be assigned*
4. *The contents of the qualified certificate on which the signature is based, and of the appropriate qualified attribute certificates, and*
5. *The results of the subsequent check of certificates under Section 5(1) Sentence 2.*

*Signature-application components shall, if necessary, also make the contents of the data to be signed or already signed sufficiently evident. The signature-code owners **should** use these signature-application components or take other suitable steps to secure qualified electronic signatures.*

The use of suitable signature-application components remains at the discretion of the signature key owners. The requirement to use suitable signature-application components is emphasized (second sentence). The formulation “should” in the third sentence makes clear, with respect to the guidelines and the various interpretation options of the Signatures Act, that the use of suitable signature-application components is not a

prerequisite for creation a qualified electronic signature. This results from the fact that it is not possible to see from an electronic signature which signature-application component was used to create it. In addition, in individual cases, "other suitable steps" can provide sufficient security.

Excerpt from the Ordinance on Electronic Signatures, Annex 1:

#### *1.1 Requirements on level of evaluation*

*The evaluation must*

*b) Cover at least evaluation level EAL 4 or E3 for technical components pursuant to Section 2 no. 10 of the Signatures Act*

*d) Cover at least evaluation level EAL 3 or E 2 for signature application components pursuant to section 2 no.11 of the Signatures Act.*

*The strength of security mechanisms assessed as "high" in the case of all products pursuant to Section I No. 1.1 a) to d) in the "E 3" and "E 2"case.*

Caution:

The annex refers to sections section 15 paragraph 5 instead of section 17 paragraph 2 & paragraphs 3 1+2. This means that there are no requirements for the evaluation level for signature-application components.

### **3.2.3 Manufacturer's Declaration**

Signature-application components will first clearly indicate the creation of a qualified electronic signature and enable the data to which the signature refers to be identified. Signature-application components are also required for checking signed data. A manufacturer's declaration is required for these signature-application components.

For the exact description, see the German Signatures Act, section 17 paragraph 4.

### **3.2.4 Validity Period of Qualified Certificates**

Excerpt from the Ordinance on Ordinance on Electronic Signatures, section 14 paragraph 3:

*The validity period of a qualified certificate shall be no longer than five years and shall not exceed the period during which the applied algorithms and related parameters remain suitable.*

New scientific knowledge or technical progress could mean that the applied algorithms and related parameters become less secure. The legislator has therefore left this backdoor open.

### 3.3 Europe-Wide Implementation of the Signature Guidelines

The European Union established common prerequisites for electronic signatures for its member states in 1999 with the Signature Guidelines. The aims of the guidelines were to be implemented from a technical point-of-view in national law in all EU countries by 19 July 2001.

The implementation has taken place in the following countries:

- Ireland 5 November 2002
- Austria 1 January 2003
- France 1 July 2003
- Germany (22 May 2001, administrative law not yet complete)

Complete by 1 January 2004:

- Belgium
- Denmark
- Finland
- Greece
- The Netherlands
- Spain
- United Kingdom
- Sweden
- Luxemburg

No regulations yet:

- Italy
- Portugal

### 3.4 ISIS-MTT

ISIS-MTT is a common specification of TeleTrusT and the T7 groups (members include DATEV, Deutsche Post Signtrust, Deutsche Telekom Telesec, TC-TrustCenter, and Sparkassen Informatik) for electronic signatures, encryption, and public-key Infrastructures. PKCS#7 signatures and signatures for S/MIME conform to ISIS-MTT. XML signatures have not yet been included in the ISIS-MTT specifications, but work is currently being done on this.

<b>Component</b>	<b>Description</b>	<b>Release</b>	<b>Transaction/Process</b>
Secure Store and Forward (SSF)	Technology interface for digital signatures	R/3 4.0A and above	
mySAP Healthcare		IS-H 4.63B with 4.6C Basis	Material requisition/Goods issue
mySAP PLM			
	Engineering Change Management (ECH)	4.6C	Status changes of engineering change orders
	Engineering Change Management (ECH)	4.6C	Status changes of object management records
	Document Management System (DMS)	4.6C	Document management: status changes
	Production Planning – Process Industries (PP-PI)	4.6C	PI sheet: complete process step
	Production Planning – Process Industries (PP-PI)	4.6C	PI sheet: accept invalid input values
	Production Planning – Process Industries (PP-PI)	4.6C	Batch record: approval
	Quality Management (QM)	4.6C	Inspection lot: results recording
	Quality Management (QM)	4.6C	Inspection lot: usage decision
	Quality Management (QM)	4.6C	Physical-sample drawing
	Cprojects (BSP application)		Digitally signing approvals
SAP Content Server		HTTP 4.5 Interface	Authenticating a request to access the archive
mySAP Public Sector	WebRequests (within CRM) can be signed digitally on the requester side (client)	Web AS 6.20 SP8, CRM 4.0	Digitally signing WebRequests
HCM Belgium	Dimona (sending employment data to	4.5B – 4.7	PC00_M12_DMNSIGNDECL

---

	social insurance)		/ Dimona digital signature
EBP	Project solutions only		
CRM	Digital signature for Web Requests	CRM4.0	Web Requests