cosign by arx

The Digital Signature Company

CoSign Connector for SAP

Version 2.4

Notice

Trademarks

ARX (Algorithmic Research) Ltd, Tel. 1-866-EASY-PKI (327-9754) Site: www.arx.com

CoSign Connector for SAP
Pub. Date 02.15

Table of Contents

Introduction

The CoSign Connector for SAP is implemented as a Windows service and it serves as a bridge between SAP and CoSign. The service makes use of the SAP .Net Connector of Microsoft. The SAP .Net Connector enables communication between the Microsoft .Net platform and SAP Systems. The CoSign Connector for SAP uses SAP Remote Function Calls (RFCs) in order to expose the CoSign Signature API Local (SAPI) to SAP (ABAP).

Once the service is properly installed and configured, SAP programmers are provided with a rich ABAP scripts signing API.

Intended Audience

This guide is intended for SAP developers looking to integrate SAP applications with CoSign Digital Signatures capabilities. It is assumed that the developer is familiar with SAP/ABAP development environment.

CoSign Connector for SAP Capabilities

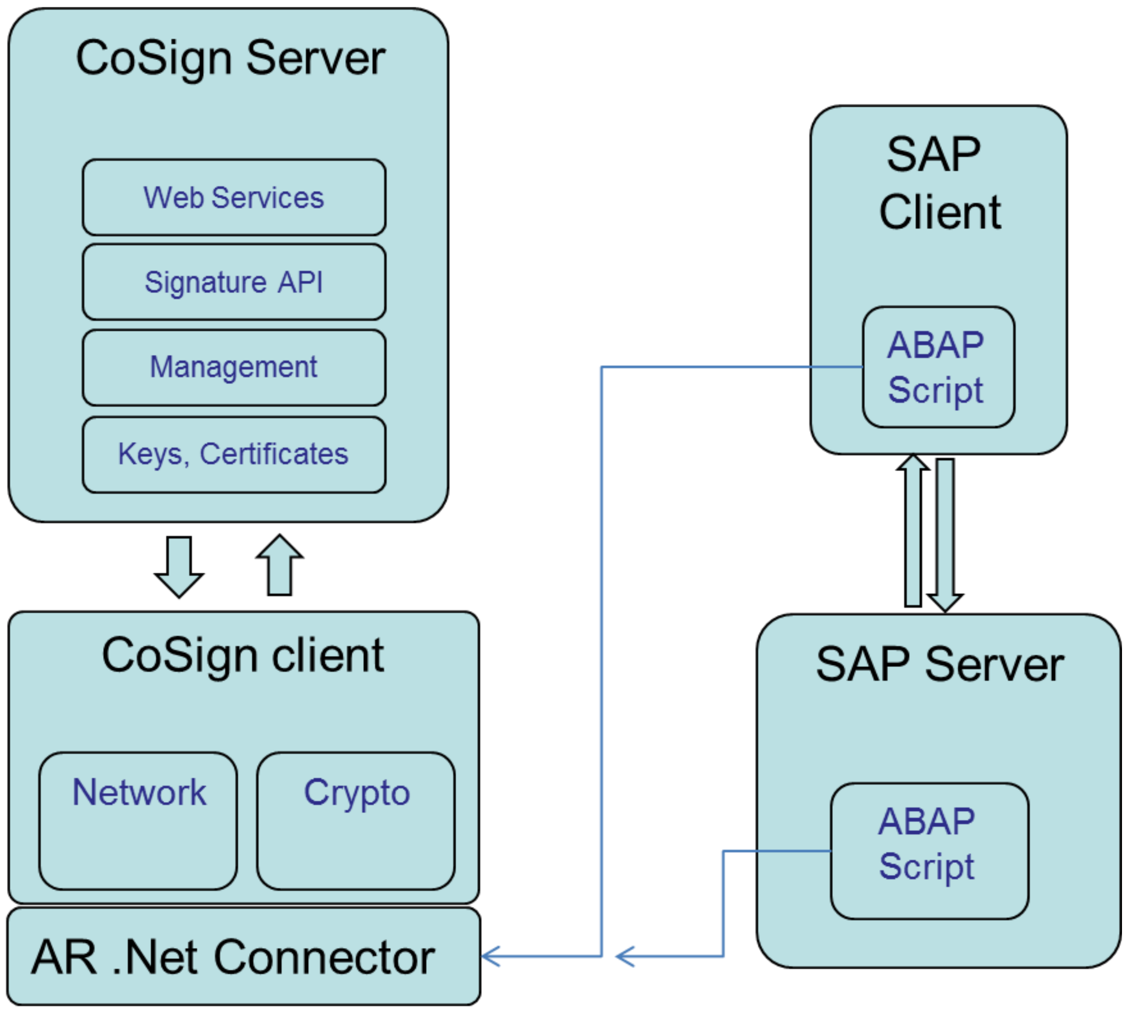
CoSign for SAP provides a single API providing:

- **Buffer signing and verification**. These methods enable signing data buffers and verifying data buffer signatures.
- **PDF signing and verification**. These methods enable signing PDF documents and verifying PDF document signatures.

Architecture of the CoSign Connector for SAP

The typical installation of the CoSign Connector for SAP consists of

- CoSign Server, either a CoSign Central appliance or CoSign Cloud
- A CoSign Client Server (CC Server). It is usually another server (typically win20xx server) in the customer's SAP environment. It hosts the CoSign Client and the CoSign Connector for SAP Windows service. Although the CoSign Client and the CoSign SAP service can reside in the SAP server itself, it is not common to do so.



Installation

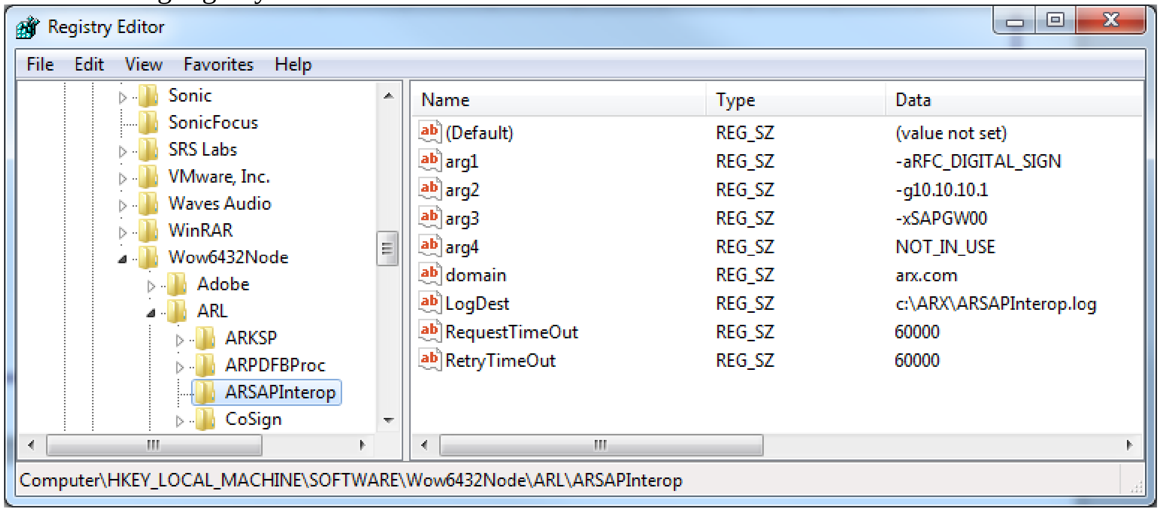
Prerequisites for the *CoSign SAP Service*

Prepare a server (typically win20xx server) in the customer's SAP environment that includes:

- .NET Framework (2.0 and later).
- Preinstalled SAP GUI (7.20 and later) application.
- Installed and configured CoSign Client. For more information regarding CoSign Client refer to *CoSign Administrator Guide.*

Installation instructions

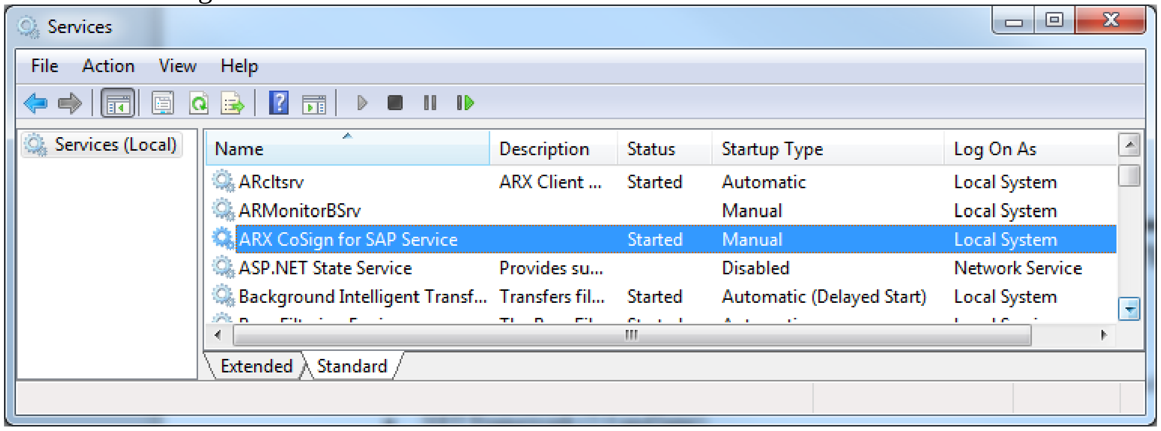
- Double click setup.exe from the installation folder and follow instructons.
- At the end of the installation you should observe:
  - The following registry entries:



Name	Type	Data
(Default)	REG_SZ	(value not set)
arg1	REG_SZ	-aRFC_DIGITAL_SIGN
arg2	REG_SZ	-g10.10.10.1
arg3	REG_SZ	-xSAPGW00
arg4	REG_SZ	NOT_IN_USE
domain	REG_SZ	arx.com
LogDest	REG_SZ	c:\ARX\ARSAPInterop.log
RequestTimeOut	REG_SZ	60000
RetryTimeOut	REG_SZ	60000

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ARL\ARSAPInterop

- The "ARX CoSign for SAP" Service:

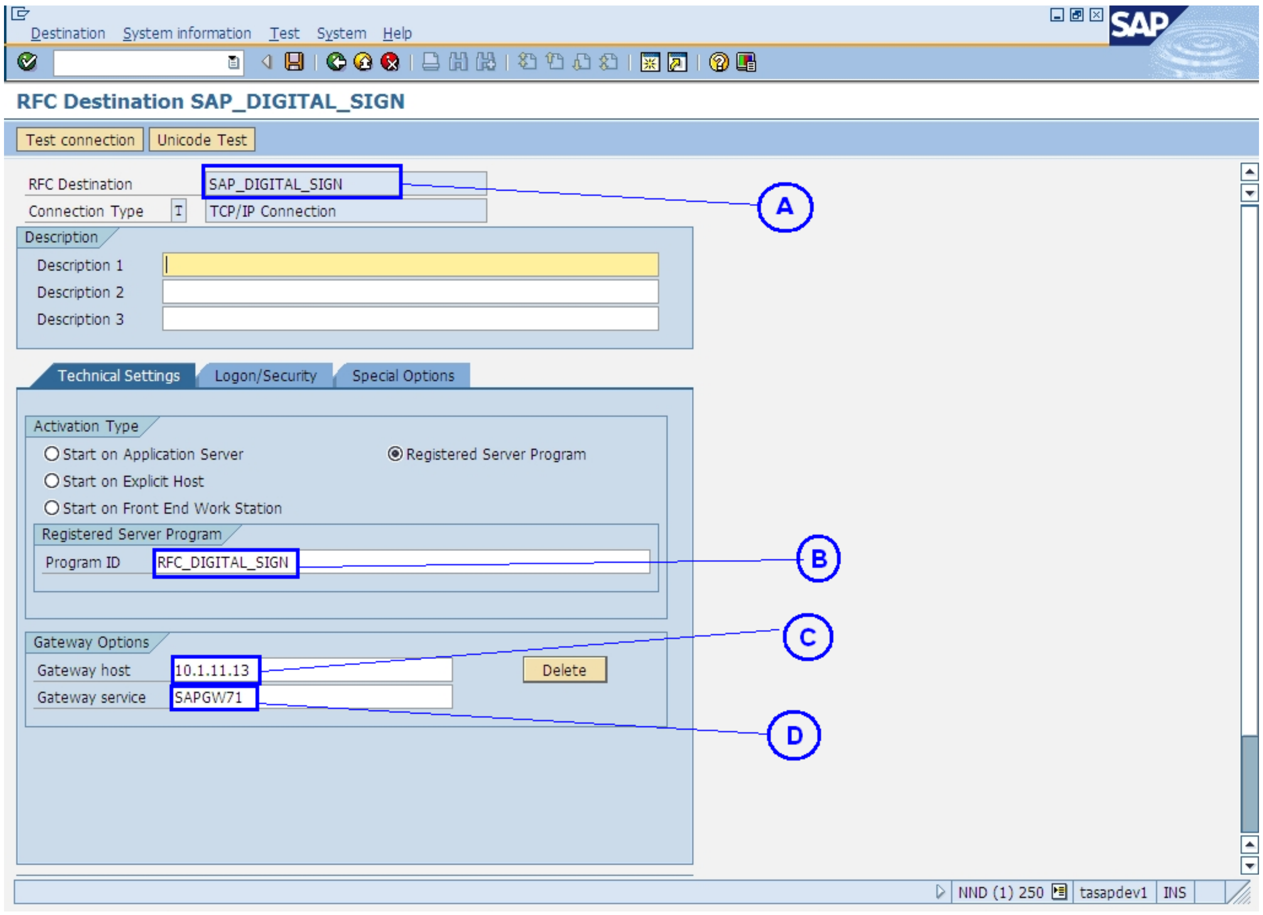


Name	Description	Status	Startup Type	Log On As
ARcltsrv	ARX Client ...	Started	Automatic	Local System
ARMonitorBSrv			Manual	Local System
ARX CoSign for SAP Service		Started	Manual	Local System
ASP.NET State Service	Provides su...		Disabled	Network Service
Background Intelligent Transf...	Transfers fil...	Started	Automatic (Delayed Start)	Local System

Post installation configuration of *CoSign SAP Service*

In order to properly use the CoSign for SAP service, please configure the following items:

- In SAP system the Transaction Code **SM59** must be configured by adding a new node under **TCP/IP** (T) connection:



RFC Destination SAP_DIGITAL_SIGN

RFC Destination: SAP_DIGITAL_SIGN (A)
Connection Type: T TCP/IP Connection

Description
Description 1:
Description 2:
Description 3:

Technical Settings | Logon/Security | Special Options

Activation Type
( ) Start on Application Server
( ) Start on Explicit Host
( ) Start on Front End Work Station
(●) Registered Server Program

Registered Server Program
Program ID: RFC_DIGITAL_SIGN (B)

Gateway Options
Gateway host: 10.1.11.13 (C)
Gateway service: SAPGW71 (D)

- ♦ A – RFC Destination name to be used in ABAP script (e.g., SAP_DIGITAL_SIGN). This SAP parameter binds the ABAP script to this SAP transaction.
- ♦ B – Program ID. This parameter must equal to arg1 of the AR SAP Service registry parameters.
- ♦ C – Gateway host. This parameter must equal to arg2 of the AR SAP Service registry parameters.
- ♦ D – Gateway service. This parameter must equal to arg3 of the AR SAP Service registry parameters.
- ♦ B,C and D parameters mark the ARX SAP Service as the implementation of this SAP transaction.

- ♦ Configure CoSign for SAP service Registry parameters:
  - ♦ The ARSapInterop service configuration parameters are taken from the Registry from "HKEY_LOCAL_MACHINE\SOFTWARE\(*Wow6432Node*)\ARL\ARSAPInterop"
  - ♦ *Wow6432Node* – present only if ARX SAP Service is installed in 64Bit windows servers.
  - ♦ arg1 – must equal to the Program ID in SM59. The parameter must be preceded by "–a" (for example –aRFC_DIGITAL_SIGN).
  - ♦ arg2 – must equal to the SAP Gateway Host in SM59. The parameter must be preceded by "–g" (for example –g10.1.11.13).
  - ♦ arg3 – must equal to the SAP Gateway Service SM59. The parameter must be preceded by "–x" (for example –xSAPGW71).
  - ♦ domain – In cases when the CoSign is installed in Active Directory or Novell environment it should be set to the domain name. In cases when the CoSign is installed in Directory Independent mode (DI) it should be set to an empty string.
  - ♦ LogDest – path to valid ARX SAP Service Log file. Make sure the path is "writable".
  - ♦ RequestTimeOut – maximal miliseconds to wait before ARX SAP Service aborts SignSC, Verify and GetChall operations.
  - ♦ RetryTimeOut – amount of miliseconds ARX SAP Service waits before restarting, when restarted from by administrator from windows services.
- ♦ After all configuration is complete, goto windows services, change the "ARX CoSign for SAP" Service "Startup mode" to Automatic and start the service.
  - ♦ The service should start correctly and you should observe log entries in the specified log file.

Buffer Signing

```
// Signs Buffer with standard Username/Domain/Password Authentication
void SignBuffer(
```

	string	UserName,	[IN]	CoSign account username
	string	Domain,	[IN]	CoSign Domain. If domain is Empty string, the value is read from: HKLM\Software\ARL\ARSAPInterop\Domain
	string	Password,	[IN]	CoSign account password
	byte[]	DataToSign,	[IN]	Buffer to Sign
	int	DataToSignLen,	[IN]	Length of the buffer to Sign
	ref String	Signature,	[OUT]	Buffer to put signature in – BASE64
	ref int	Result);	[OUT]	Return code

```
// Sign buffer with stanadrd Username/Domain/Password authentication while
// providing a signature password for “prompt for sign” mode.
void SignBufferEx(
```

	string	UserName,	[IN]	CoSign account username
	string	Domain,	[IN]	CoSign Domain. If domain is Empty string, the value is read from: HKLM\Software\ARL\ARSAPInterop\Domain
	string	Password,	[IN]	CoSign account password
	string	SigPassword,	[IN]	CoSign signature password
	byte[]	DataToSign,	[IN]	Buffer to Sign
	int	DataToSignLen,	[IN]	Length of the buffer to Sign
	ref String	Signature,	[OUT]	Buffer to put signature in – BASE64
	ref int	Result);	[OUT]	Return code

```
// Signs Buffer with smartcard authentication
void SignBufferSC(
```

	string	UPN,	[IN]	CoSign account User Principle Name
	string	Domain,	[IN]	CoSign Domain. If domain is Empty string, the value is read from: HKLM\Software\ARL\ARSAPInterop\Domain
	string	Password,	[IN]	CoSign account password
	string	SignedB64Chall,	[IN]	Sign challenge in BASE64 format
	int	UseSCForLogon,	[IN]	Indication if the authentication should be used for logon operation
	byte[]	DataToSign,	[IN]	Buffer to Sign
	int	DataToSignLen,	[IN]	Length of the buffer to Sign
	byte[]	Signature,	[OUT]	Buffer to put signature – binary
	int	SignatureLen,	[OUT]	Length of returning signature
	ref int	Result);	[OUT]	Return code

```
//Verifies Signed Buffer
void VerifyBuffer(
```

	byte[]	Buffer,	[IN]	Buffer containing data that was signed
	int	BufferLen,	[IN]	Length of the Buffer
	String	Signature,	[IN]	String in BASE64 containing the signature
	ref int	isValid,	[OUT]	1 – if signature valid, 0 – otherwise
	ref string	Signer,	[OUT]	Signer details
	ref int	Result);	[OUT]	Return code

PDF Signing

```
// Signs PDF document with standard Username/Domain/Password Authentication
void SignPDF(
```

	string	UserName,	[IN]	CoSign account username
	string	Domain,	[IN]	CoSign Domain. If domain is Empty string, the value is read from: HKLM\Software\ARL\ARSAPInterop\Domain
	string	Password,	[IN]	CoSign account password
	string	FileName,	[IN]	Complete path to PDF file to Sign
	int	Invisible,	[IN]	0 – for creating invisible signature, 1 – for visible
	int	page,	[IN]	the page number
	int	x,	[IN]	signature field X-position
	int	y,	[IN]	signature field Y-position
	int	height,	[IN]	signature field height
	int	width,	[IN]	signature field width
	string	Reason,	[IN]	Reason for signing
	int	isDisplayGraphSig,	[IN]	1/0, sets if Graph Sig element should appear
	int	isDisplayUsername,	[IN]	1/0, sets if Username element should appear
	int	isDisplayDateTime,	[IN]	1/0, sets if Date/Time element should appear
	ref int	Result);	[OUT]	Return Code

```
// Sign PDF with stanadrd Username/Domain/Password authentication while
// providing a signature password for “prompt for sign” mode
void SignPDFEx(
```

	string	UserName,	[IN]	CoSign account username
	string	Domain,	[IN]	CoSign Domain. If domain is Empty string, the value is read from: HKLM\Software\ARL\ARSAPInterop\Domain
	string	Password,	[IN]	CoSign account password
	string	SigPassword,	[IN]	CoSign signature password
	string	FileName,	[IN]	Complete path to PDF file to Sign
	int	Invisible,	[IN]	0 – for creating invisible signature, 1 – for visible
	int	page,	[IN]	the page number
	int	x,	[IN]	signature field X-position
	int	y,	[IN]	signature field Y-position
	int	height,	[IN]	signature field height
	int	width,	[IN]	signature field width
	string	Reason,	[IN]	Reason for signing
	int	isDisplayGraphSig,	[IN]	1/0, sets if Graph Sig element should appear
	int	isDisplayUsername,	[IN]	1/0, sets if Username element should appear
	int	isDisplayDateTime,	[IN]	1/0, sets if Date/Time element should appear
	ref int	Result);	[OUT]	Return Code

```
// Signs PDF with SmartCard Authentication
void **SignPDFSC**(
```

	string	UserName,	[IN]	CoSign account username
	string	Domain,	[IN]	CoSign Domain. If domain is Empty string, the value is read from: HKLM\Software\ARL\ARSAPInterop\Domain
	string	Password,	[IN]	CoSign account password
	string	SignedB64Chall,	[IN]	Base64-encoded ticket for SmartCard Auth
	int	isUseSCforLogon,	[IN]	0 – if SC-Auth is required for the Sign() operation only, 1 – if SC-Auth is required also for Logon(). If this parameter is set to 1, the Password parameter value will be ignored.
	string	FileName,	[IN]	Complete path to PDF file to Sign
	int	Invisible,	[IN]	0 – for creating invisible signature, 1 – for visible
	int	page,	[IN]	the page number
	int	x,	[IN]	signature field X-position
	int	y,	[IN]	signature field Y-position
	int	height,	[IN]	signature field height
	int	width,	[IN]	signature field width
	ref int	Result);	[OUT]	Return Code

```
void **VerifyPDF**(
```

	string	FileName,	[IN]	Complete path to PDF file to Verify
	ref int	SignaturesStatus,	[OUT]	-1 – Error encountered. Check Result for error code. 0 - all fields are signed, all signatures valid 1 - some fields are signed, all signatures valid 2 - some fields are signed, some signatures invalid 3 - there are no signatures on the document (there might be fields)
	ref string	Signer,	[OUT]	Not in use.
	ref int	Result);	[OUT]	Return code

ABAP script call to AR SAP Service’s SignPDF() - sample

```
CALL FUNCTION ‘SignPDF’ DESTINATION ‘SAP_DIGITAL_SIGN’
   EXPORTING
      Username            =  ‘John Miller’
      Password            =  ‘12345678’
      Filename            =  ‘C:\Temp\Invoice.pdf’
      Invisible           =  0                       “or 1
      Page                =  1
      X                   =  94
      Y                   =  126
      Height              =  159
      Width               =  159
      Reason              =  ‘I approve’
      Is_Display_Graph    =  1                       “or 0
      Is_Display_Username =  1                       “or 0
      Is_Display_DateTime =  1                       “or 0
   IMPORTING
      RESULT = result
   EXCEPTIONS
      nothing_specified    =  1
      no_record_found      =  2
      OTHERS               =  3.
```