TempelPlus User Manual

Software version: 1.0.0
Date: 31.08.2011

Sertifitseerimiskeskus AS

2011

Table of contents

1 Introduction

The TempelPlus software is designed for signing of large quantities of files (i.e. mass signing) with **the Digital stamps of the institution** (http://www.sk.ee/en/services/Digital-stamp), for instance for signing of invoices, payment orders etc. The TempelPlus software is convenient when there are many files to be signed and a separate signed container (.ddoc file) must be created for each file. The Digital stamp of the institution can also be used with the DigiDoc Client software but that solution is suitable only for creating one signed container at a time. The advantage of the TempelPlus software is that the PIN must only be entered once for multiple signatures.

Besides signing, TempelPlus also performs mass encryption and decryption, verification of large quantities of .ddoc files and so on.

2 Version information

2.1 Version 1.0.0, release date 31.08.2011

Tested in Windows 7 using a Digital stamp on the SafeNet Aladdin eToken Pro USB token and Digital stamp on smart card. This software version is a command line-only utility for use in Windows operating systems.

- Improved: command line utility help information
- Improved: working with TempelPlus command line utility in Windows no longer requires administrator rights
- Added: **–signer_cn** parameter for the signing function (**tempelplus sign**), enabling selection during signing of a particular key from those on the signing token.

2.2 Version 0.99 (beta), release date 27.12.2010

First public beta version. This software version is a **command line-only** utility, tested in the Windows 7 operating system. Available as a Windows installation package (.msi file). The primary functions of TempelPlus – signing and encryption/decryption – have been tested with the following tokens:

- On a smart card, institution ID (Digital stamp), with signing and authentication/encryption certificates, 1024 (1K) bit keys
- On a USB token (Aladdin (SafeNet) eToken Pro), institution ID (Digital stamp), with signing and authentication/encryption certificates, 1K and 2 K keys, with only 1 PIN (i.e. the USB token does not have secondary authentication passwords for using the keys saved there).

3 Installation

The software is installed from the Windows MSI installation package downloaded from http://www.sk.ee/tempelplus. The installation package will create the TempelPlus software group in All Programs, with all the shortcuts needed to work with TempelPlus.

4 Uninstallation

To uninstall TempelPlus in Windows: All Programs → TempelPlus → uninstall.

5 Help

Feel free to ask questions and submit TempelPlus software improvement suggestions to SK customer support: abi@id.ee.

6 Setup

The TempelPlus software is based on the JDigiDoc library/utility (see http://www.id.ee/28729?id=28733). TempelPlus therefore contains both its own settings and JDigiDoc library settings. The default TempelPlus settings are made during installation of the software so that TempelPlus is ready for use immediately after installation. The settings can also be manually changed if/when necessary.

6.1 JDigiDoc settings

The most important settings of the JDigiDoc library/utility are located in the following files:

- **<TempelPlus home directory>\JDigiDoc\jdigidoc.bat**, primary parameters, set during TempelPlus installation:
    - **JDIGIDOC_HOME** – the JDigiDoc library location in the file system
    - **JAVA_PATH** – the java.exe file location in the file system, should be changed only if the installation package cannot find Java.
- **<TempelPlus home directory>\JDigiDoc\jdigidoc-win.cfg**, JDigiDoc library settings, no need to change for using TempelPlus.

More information about the JDigiDoc library/utility, including its configuration, is available in the JDigiDoc documentation: **<TempelPlus home directory>\JDigiDoc\doc**

6.2 TempelPlus settings

The most important TempelPlus settings are located in the following files:

- **<TempelPlus home directory>\TempelPlus.bat**
    - Parameter **JAVA_PATH** – the java.exe file location in the file system, should be changed only if the installation package cannot find Java
- **<TempelPlus home directory>\SignatureLogging.properties**
    - Parameter **log4j.appender.file.file** – the TempelPlus log file location in the file system, determined by the installation package
- **<TempelPlus home directory>\TempelPlus.conf**
    - *Parameters connected with signing, encryption*

- **role**, **country**, **state**, **city**, **postcode** – the parameters determining the signer role, location
- **format** – the format of the created signed containers, the sole currently valid option is: ddoc
- **crypt** – the file extension of the encrypted container, the default value is: cdoc
- **pin_enter** – the PIN query manner: graphic or command line, the corresponding options are: graphic, console
- **DIGIDOC_OCSP_RESPONDER_URL** – the Online Certificate Status Protocol (OCSP) service address; Live service URL: http://ocsp.sk.ee; Test service URL: http://www.openxades.org/cgi-bin/ocsp.cgi
- **SIGN_OCSP_REQUESTS** – whether the Online Certificate Status Protocol (OCSP) service queries are signed (in case of access token (a PKCS#12 container)) or not (in case of IP address-based access), these are the possible options: true, false
- **DIGIDOC_PKCS12_CONTAINER** – the access token file path, used with parameters **SIGN_OCSP_REQUESTS=true** and **DIGIDOC_PKCS12_PASSWD**
- **DIGIDOC_PKCS12_PASSWD** – the access token password, used with parameters **SIGN_OCSP_REQUESTS=true** and **DIGIDOC_PKCS12_CONTAINER**

- *Other settings – files, directories*
  - **jddoc_location** – the JDigiDoc library location, determined by the installation package
  - **log_file** – the log file location, determined by the installation package
  - **work_directory** – the TempelPlus work directory, for temporary files, etc., determined by the installation package
  - **DIGIDOC_LOG4J_CONFIG** – the SignatureLogging.properties file location, determined by the installation package
  - **DIGIDOC_DF_CACHE_DIR** – the JDigiDoc library's directory for temporary files, determined by the installation package
  - **bc_prov=bcprov-jdk15-125.jar** – this preset parameter should not be changed

- **DIGIDOC_MAX_DATAFILE_CACHED=1000** – this preset parameter should not be changed
- **DIGIDOC_SIGN_PKCS11_DRIVER** – the PKCS#11 driver location in the file system. Determined by the installation package and if a driver is found, the priority sequence is as follows:
  - Aladdin eToken driver (C:\\windows\\system32\\eTPKCS11.dll)
  - OpenSC PKCS#11 driver included in ID card software (C:\\Windows\\System32\\opensc-pkcs11.dll)

- *Other settings – application behaviour*
  - **control_question** – whether the user is asked control questions to confirm actions, the options are: yes/no
  - **date_format** – the date display format, the default value is: dd.MM.yyyy HH:mm:ss

7 Using the software – by usecases

7.1 Using software help

Open the command line and navigate to the TempelPlus home directory (replace with the suitable directory if necessary):
cd "c:\Program Files\TempelPlus"

General help about using TempelPlus is accessible with this command:
tempelplus –help or
tempelplus -?

Help about a particular command, for instance signing, is accessible with this command:
tempelplus sign –help or
tempelplus sign -?

7.2 Signing

You can sign one file or a whole directory (folder) of files simultaneously. The files created after signing can be saved in the same folder as the source files or you can choose another folder (the latter option is recommended). The source files that are being signed can be deleted during the signing process. It is also possible to leave the application running in the background (standby mode): after the source files are signed, the application will remain open and wait for new files to appear for signing in the input folder. The general principle is that if a new file is created in the output directory for signed files where a file of the same name already exists, the file will not be overwritten but the software will close itself with an error message. If the file to be signed is already a DigiDoc container, one new signature will be added to it during the process. If the file to be signed is not a DigiDoc container, the software will create the corresponding DigiDoc container with one signature. **NB! When signing files in TempelPlus, connect the Digital stamp to your computer. If the Digital stamp is on a smart card, just insert it in the card reader. If the Digital stamp is on a USB token, plug it into your computer's USB port.**

Open the command line and navigate to the TempelPlus home directory (replace with the suitable directory if necessary):
cd "c:\Program Files\TempelPlus"

The signing command generally looks like this:
tempelplus sign <filer or folder to be signed> <additional parameters>

Additional parameters:

- **-output_folder <folder>** – this optional parameter, if set, determines the folder where the signed files are saved
- **-remove_input** – this optional parameter, if set, results in deletion of the source files
- **-follow** – this optional parameter, if set, puts the application in the standby (or continuously running) mode. **NB!** This parameter must be used with these parameters: **–remove_input** and **–output_folder**
- **-signer_cn "<CN>"** – this optional parameter, if set, determines the key to be used for signing. If the parameter is not used, the first signing key on signing token is selected. The parameter value, if set, must be the value of the Subject CN (Common Name) field of the certificate associated with the signing key.
- **-role "<role>"** – this optional parameter, if set, enables addition of the signer role to the signed container. If the signer **role** is also set in the TempelPlus.conf configuration file, the preference is given to the command parameter
- **-country "<country>"** – this optional parameter, if set, enables addition of the signer location information to the signed container. If the **country** parameter is also set in the TempelPlus.conf configuration file, the preference is given to the command parameter

- **-state "<state/county>"** – this optional parameter, if set, enables addition of the signer location information to the signed container. If the **state** parameter is also set in the TempelPlus.conf configuration file, the preference is given to the command parameter

- **-city "<city>"** – this optional parameter, if set, enables addition of the signer location information to the signed container. If the **city** parameter is also set in the TempelPlus.conf configuration file, the preference is given to the command parameter

- **-postcode "<postcode>"** – this optional parameter, if set, enables addition of the signer location information to the signed container. If the **postcode** parameter is also set in the TempelPlus.conf configuration file, the preference is given to the command parameter

Example: Signing setting folder containing source files (for instance: C:\input\) and output folder (C:\output\). One signed DigiDoc container is created in the output folder for each file in the input folder:

tempelplus sign c:\input -output_folder c:\output

The software will ask the user to enter the PIN only once for signing of a file folder, it will then display in real time the file number being currently signed and if the process ends successfully, the user will see a notification similar to this:

Done
7 documents signed successfully
TempelPlus v0.99 stopping. Time used: 18 seconds

7.3 Signature verification

The TempelPlus software allows the user to verify signature validity of a whole folder of signed files. The folder can contain both DigiDoc containers and ordinary data files.

Open the command line and navigate to the TempelPlus home directory (replace with the suitable directory if necessary):
cd "c:\Program Files\TempelPlus"

The signature verification command generally looks like this:
tempelplus verify <file or folder to be verified >

Example: verification of the signatures in the signed containers in one folder (for instance: C:\digidoc_files\):

tempelplus verify c:\digidoc_files

The software will display the file number currently being processed in real time, it will also display information about the found signature – signer, date, validity – and if the process ends successfully, the user will see a notification similar to this:

Done
7 documents verified successfully
TempelPlus found 14 valid signatures and 0 invalid signatures
TempelPlus v0.99 stopping. Time used: 2 seconds

7.4 Signature removal

TempelPlus enables simultaneous removal of signatures from many files in two ways: either all signatures are removed from all signed files in the selected folder or only a particular person's signatures are removed from those signed files. The result will be in the form of DigiDoc files with one or all signatures removed.

Open the command line and navigate to the TempelPlus home directory (replace with the suitable directory if necessary):
cd "c:\Program Files\TempelPlus"

The signature removal command generally looks like this:
tempelplus remove <signature identifier> <source file or input folder> -output_folder <output folder>

These are the possible signature identifiers:

- **ALL** – to remove all signatures in the signed file
- **"<CN>"** – to remove from the signed file only a signature added there by a particular individual/institution. The person is set by the value of the Subject CN (Common Name) field of the signing certificate associated with the Digital stamp or that person's ID card. In case of signatures given by ID card, it will look like this: "Surname,First name,Personal identification code"

Example 1: the user wants to remove all signatures from all signed containers in one folder (for instance: C:\digidoc_files\) and save the DigiDoc containers stripped of the signatures in another folder (for instance: C:\digidoc_files2\):

tempelplus remove ALL c:\digidoc_files\ -output_folder c:\digidoc_files2

The software will display the file number currently being processed in real time, it will also display signer information for the found signatures and if the process ends successfully, the user will see a notification similar to this:

Done
7 documents were handled successfully. 14 signatures removed
TempelPlus v0.99 stopping. Time used: 4 seconds

Example 2: the user wants to remove from all signed containers in one folder (for instance: C:\digidoc_files\) the signatures of one particular person (MARI-LIIS MÄNNIK, 47101010033) and save the DigiDoc containers without these signatures in another folder (for instance: C:\digidoc_files2\):

tempelplus remove "MÄNNIK,MARI-LIIS,47101010033" c:\digidoc_files\ -output_folder c:\digidoc_files2

The software will display the file number currently being processed in real time, it will also display signer information for the found signatures and if the process ends successfully, the user will see a notification similar to this:

Done
7 documents were handled successfully. 2 signatures removed
TempelPlus v0.99 stopping. Time used: 2 seconds

7.5 Extraction of data files from container

With TempelPlus you can extract the data files from the (signed) digidoc containers located in a particular folder. A folder of the same name as the container (for instance: fail1.ddoc) will be created to save the data files extracted from the container.

Open the command line and navigate to the TempelPlus home directory (replace with the suitable directory if necessary):
cd "c:\Program Files\TempelPlus"

The command for data file extraction generally looks like this:
tempelplus extract <source file or input folder> -output_folder <output folder>

Example 1: the user wishes to extract all data files from all signed containers in one folder (for instance: C:\digidoc_files\) and use another folder (for instance: C:\digidoc_files2\) to create subfolders named after the corresponding DigiDoc containers, with each subfolder containing the data files from a particular container:

tempelplus extract c:\digidoc_files\ -output_folder c:\digidoc_files2

The software will display the file number currently being processed in real time and if the process ends successfully, the user will see a notification similar to this:

Done
2 documents where handled successfully. 3 files extracted
TempelPlus v0.99 stopping. Time used: 2 seconds

7.6 Creating containers containing several data files

TempelPlus allows the user to simultaneously create many DigiDoc containers (without signatures) containing one or several data files.

Open the command line and navigate to the TempelPlus home directory (replace with the suitable directory if necessary):
cd "c:\Program Files\TempelPlus"

The command for container creation generally looks like this:
tempelplus container <source file or input folder> <additional parameters>

If a source file is given, the software will create one container. If an input folder with several files is given, the software will create as many containers as there are files in the input folder, i.e. one DigiDoc container for each file.

Additional parameters:

- **-output_folder <folder>** – this optional parameter, if set, determines the folder to which the created containers are saved
- **-add_file <folder or list of files with blank space separation>** – this optional parameter, if set, enables addition of the designated files to every created container

Example 1: the input folder (for instance: C:\datafiles\) has many data files. The user wishes to use another folder (for instance: C:\digidoc_files) to save one DigiDoc container for each data file in the input folder, with that container comprising the corresponding data file from the input folder:

tempelplus container c:\datafiles\ -output_folder c:\digidoc_files

The software will display the file number currently being created in real time and if the process ends successfully, the user will see a notification similar to this:

Done
7 documents created successfully.
TempelPlus v0.99 stopping. Time used: 1 seconds

Example 2: the input folder (for instance: C:\datafiles\) has many data files. The user wishes to use another folder (for instance: C:\digidoc_files) to save one DigiDoc container for each data file in the input folder, with that container comprising the corresponding data file from the input folder and 2 additional files:

tempelplus container c:\datafiles\ -output_folder c:\digidoc_files\ -add_file c:\file1.txt c:\file2.dat

The software will display the file number currently being created in real time and if the process ends successfully, the user will see a notification similar to this:

Done
7 documents created successfully.
TempelPlus v0.99 stopping. Time used: 1 seconds

Example 3: the input folder (for instance: C:\datafiles\) has many data files. The user wishes to use another folder (for instance: C:\digidoc_files) to save one DigiDoc container for each data file in the input folder, with that container comprising the corresponding data file from the input folder and all files from the folder C:\datafiles2:

tempelplus container c:\datafiles\ -output_folder c:\digidoc_files\ -add_file c:\datafiles2

The software will display the file number currently being created in real time and if the process ends successfully, the user will see a notification similar to this:

Done
7 documents created successfully.
TempelPlus v0.99 stopping. Time used: 1 seconds

7.7 Encryption

With TempelPlus you can encrypt a whole folder of files for one or several recipients, i.e. an individual or institution that can then decrypt the encrypted files using their ID card or Digital stamp.

Open the command line and navigate to the TempelPlus home directory (replace with the suitable directory if necessary):
cd "c:\Program Files\TempelPlus"

The encryption command generally looks like this:
tempelplus encrypt <source file or input folder> -cert <one or several certificate files> -output_folder <output folder>

If the encryption output is for several recipients, the certificate files must have blank space separation. **–output_folder** is an optional parameter.

Example: the input folder (for instance: C:\datafiles\) has many data files. The user wishes to encrypt the files for two individuals/institutions (on the condition that the user has the certificate files for the corresponding ID card/Digital stamp authentication (encryption)) and save the encrypted containers in a particular output folder (for instance: C:\encrypted_files):

tempelplus encrypt c:\datafiles\ -cert c:\certs\person1_auth.cer c:\certs\institution2_crypt.cer -output_folder c:\encrypted_files

The software will display the file number currently being created in real time and if the process ends successfully, the user will see a notification similar to this:

Done
7 files encrypted successfully!
TempelPlus v0.99 stopping. Time used: 5 seconds

7.8 Decryption

TempelPlus allows the user to decrypt a whole folder of encrypted files on the condition that the recipient of the encrypted files is the owner of the Digital stamp that must be used for decryption. So for the user to be able to decrypt the received files with the Digital stamp, the files must have been encrypted using the authentication (encryption) certificate of the same Digital stamp. For each encrypted container a folder named after the encrypted file is created during the decryption process and in that folder are saved all data files from the container. **NB! When decrypting files in TempelPlus, connect the Digital stamp to your computer. If the Digital stamp is on a smart card, just insert it in the card reader. If the Digital stamp is on a USB token, plug it into your computer's USB port.**

Open the command line and navigate to the TempelPlus home directory (replace with the suitable directory if necessary):
cd "c:\Program Files\TempelPlus"

The decryption command generally looks like this:
tempelplus decrypt <source file or input folder> -recipient <CN> <additional parameters>

- **-recipient "<CN>"** – this optional parameter, if set, determines the decryption recipient. The recipient is set by the value of the Subject CN (Common Name) field of the authentication (encryption) certificate associated with the Digital stamp or that person's ID card

Additional parameters:

- **-output_folder <folder>** – this optional parameter, if set, determines the folder where the decrypted files are saved
- **-remove_input** – this optional parameter, if set, results in deletion of the (encrypted) source files
- **-follow** – this optional parameter, if set, puts the application in the standby (or continuously running) mode. **NB!** This parameter must be used with these parameters: **–remove_input** and **–output_folder**

Example: the input folder (for instance: C:\encrypted_files\) has many encrypted files. The decryption process will be performed using the Digital stamp where the authentication (encryption) certificate has this CN field value: "InstitutionX: approval of contracts". The contents of the encrypted files are saved in the output folder (for instance: C:\decrypted_files):

tempelplus decrypt c:\encrypted_files\ -output_folder c:\decrypted_files\ -recipient "InstitutionX: approval of contracts"

The software will request the user to enter the PIN (if the Digital stamp in on a smart card – PIN2, if the Digital stamp is on an Aladdin eToken USB token – PIN), then it will display the file number currently being processed in real time and if the process ends successfully, the user will see a notification similar to this:

Done
7 files decrypted successfully! 7 files created.
TempelPlus v0.99 stopping. Time used: 19 seconds