Contents

1 Introduction .................................................................................................................. 6
  1.1 Description .............................................................................................................. 6
  1.2 Functions ............................................................................................................... 6
  1.2.1 Features .......................................................................................................... 6
  1.2.2 Formats ........................................................................................................... 7
    Input Formats ........................................................................................................... 7
    Output Formats ....................................................................................................... 8
  1.2.3 Conformance .................................................................................................... 8
  1.3 Service .................................................................................................................. 8
  1.4 Operating Systems ............................................................................................... 8
  1.5 Digital Signatures .................................................................................................. 8
    1.5.1 Overview ....................................................................................................... 8
    1.5.2 Terminology .................................................................................................. 9
    1.5.3 Why Digitally Signing? ................................................................................ 9
    1.5.4 What is an Electronic Signature? .................................................................. 10
      Simple Electronic Signature ............................................................................. 10
      Advanced Electronic Signature ....................................................................... 11
      Qualified Electronic Signature ....................................................................... 11
    1.5.5 How to Create Electronic Signatures .......................................................... 11
      Preparation Steps ............................................................................................... 11
      Application of the Signature ........................................................................... 12

2 Installation ................................................................................................................... 14
  2.1 Overview ............................................................................................................... 14
  2.2 Windows ............................................................................................................... 14
  2.3 Uninstall ............................................................................................................... 14
  2.4 Note about the Evaluation License ........................................................................ 15
  2.5 Special Directories ............................................................................................... 15
    2.5.1 Directory for temporary files ....................................................................... 15
    2.5.2 Cache Directory ......................................................................................... 15
    2.5.3 Font Directories ......................................................................................... 15

3 License Management .................................................................................................. 16
  3.1 License Features ................................................................................................... 16
  3.2 License Installation and Management ................................................................. 16
    3.2.1 Graphical License Manager Tool .................................................................. 16
      List all installed license keys ............................................................................ 16
      Add and delete license keys ............................................................................ 17
      Display the properties of a license .................................................................. 17
    3.2.2 Command Line License Manager Tool ......................................................... 17
      List all installed license keys ............................................................................ 17
      Add and delete license keys ............................................................................ 17
      Display the properties of a license .................................................................. 18
  3.3 License Selection and Precedence ......................................................................... 18
    3.3.1 Selection ....................................................................................................... 18
    3.3.2 Precedence ................................................................................................... 19
  3.4 Key Update ........................................................................................................... 19
  3.5 License activation ................................................................................................. 19
    3.5.1 Activation .................................................................................................... 19
Reactivation

Connection to the licensing service fails

Proxy Setting

Deactivation

Offline Usage

First Step: Create a Request File

Second Step: Use Form on Website

Third Step: Apply the Response File

License Key Versions

License Key Storage

Windows

Troubleshooting

License key cannot be installed

License is not visible in license manager

License is not found at runtime

Eval watermark is displayed where it should not

Activation is not recognized

Activation is invalidated too often

Connection to the licensing service fails

Offline usage fails due to a request/response mismatch

Getting Started

Configuration

Retrieve Information about Available Options and Settings

Managing the Service

State Diagram of the Service

Using the Service

Log Files

User’s Guide

Process Description

Conversion Steps

Conversion Errors

Handling Conversion Errors

Post Analysis

What is PDF/A?

PDF/A-1

What is the difference between PDF/A-1b and PDF/A-1a?

PDF/A-2

PDF/A-3

Color Spaces

Colors in PDF

ICC Color Profiles

PDF/A Requirements

Fonts

Font Cache

Font Configuration File fonts.ini

Cryptographic Provider

PKCS#11 Provider

Configuration

Interoperability Support

Selecting a Certificate for Signing

Using PKCS#11 stores with missing issuer certificates

Cryptographic Suites
6.4 Color Profiles ................................................................. 73
6.4.1 -cs ICC Profile for Device-Specific Color Spaces ................. 73
6.4.2 -oi ICC Profile for Output Intent .................................. 74
6.5 Digital Signatures .......................................................... 74
6.5.1 -abg Signature Background Image ................................ 74
6.5.2 -af1 Signature Font Name 1 ......................................... 74
6.5.3 -af2 Signature Font Name 2 ........................................ 74
6.5.4 -afs1 Signature Font Size 1 ........................................ 75
6.5.5 -afs2 Signature Font Size 2 ........................................ 75
6.5.6 -ap Signature Page Number ........................................ 75
6.5.7 -ar Signature Annotation Rectangle ............................ 75
6.5.8 -at1 Signature Text 1 ................................................ 75
6.5.9 -at2 Signature Text 2 ................................................ 76
6.5.10 -ccil Signer contact info ........................................... 76
6.5.11 -cpf Certificate Fingerprint ...................................... 76
6.5.12 -ci Certificate Issuer ................................................ 76
6.5.13 -cn Certificate Name (Subject) .................................. 76
6.5.14 -cno Certificate Serial Number .................................. 77
6.5.15 -co Do not Embed Revocation Information ..................... 77
6.5.16 -cp Cryptographic Provider ...................................... 77
6.5.17 -cpf Cryptographic session property (file) ..................... 78
6.5.18 -cps Cryptographic session property (string) .................. 78
6.5.19 -cr Signature Reason ............................................... 78
6.5.20 -cs1 Certificate Store Location .................................. 79
6.5.21 -csn Certificate Store Name ...................................... 79
6.5.22 -nc Disable cache for CRL and OCSP .......................... 79
6.5.23 -st Set Signature SubFilter ....................................... 80
6.5.24 -tsc Time-stamp Credentials .................................... 80
6.5.25 -tsu Time-stamp URL .............................................. 80
1 Introduction

1.1 Description

The 3-Heights™ PDF to PDF/A Converter Service converts PDF files into PDF/A files. PDF/A has been acknowledged world-wide as the ISO standard for long-term archiving since 2005. The tool analyzes and converts the input file, applying a digital signature where required.

The integrated validator then optionally checks conformity once again. This product is robust and powerful and therefore predestined for archive migrations of any size.

1.2 Functions

The 3-Heights™ PDF to PDF/A Converter Service accepts files from many different applications and automatically converts them into PDF/A. The level of conformity can be set to level A, U, or B. ICC color profiles for device-dependent color profiles and font types are embedded in the document. There is an option to provide the entire character set for fonts (no subsetting) to facilitate editing at a later stage. Missing fonts are reproduced as close to the original as possible via font recognition. Metadata can be generated automatically or added from external sources. The tool also detects and automatically repairs problems typical of the PDF format. A digital signature can be applied and a conformity check carried out at the end of the process. The optional OCR Add-On and linearization for fast web display are valuable additional functions.

1.2.1 Features

- Convert PDF documents to PDF/A-1, PDF/A-2, PDF/A-3
- Support for all PDF/A conformance levels
- Make color spaces device-independent, e.g. by embedding ICC profile or setting an output intent
- Embed and subset fonts
- Colorants management (PDF/A-2 and later)
- Recover corrupt documents
- Repair corrupt data such as embedded font programs or images
- Remove transparency (PDF/A-1 only)
- Remove malicious content such as attached files (PDF/A-1 and PDF/A-2) and JavaScript actions
- Remove multimedia content such as video and sound
- Conversion of embedded and attached files (PDF/A-2 and later)
- Repair metadata and make them consistent

Conversion process control
- Pre- and post-validation
- Conversion reporting
- Write the application log to a log file and log to the event log of the operating system
- Automatically determine optimal conformance based on input file (optional)
- Enables sophisticated error handling

Digital signatures, conforming to PDF/A
- Apply PAdES-LTV (Long Term Validation) signatures
- Embedded trust chain, time-stamp and revocation information (OCSP, CRL)
- Various types of cryptographic providers
  - Windows certificate store
  - Hardware such as hardware security module (HSM), smart cards, and USB tokens
  - Online signature services
    - SwissSign Digital Signing Service
    - Swisscom All-in Signing Service
    - GlobalSign Digital Signing Service
    - QuoVadis sealsign
  - Add an optional visual appearance of the signature (page, size, color, position, text, background image, etc.)

- Read encrypted input files
- Enhance output file
  - Set metadata
  - Linearization for fast web view
  - Use PDF file compression features (PDF/A-2 and later)
- Text recognition using OCR engine (optional)
  - Replace old OCR text or skip images with existing OCR text
  - Set the OCR language and options
  - Deskew and de-noise images
  - Detect barcodes
  - List OCR plug-ins

1.2.2 Formats

Input Formats
- PDF 1.x (PDF 1.0, . . ., PDF 1.7)
- PDF 2.0
Output Formats

- PDF/A-1a, PDF/A-1b
- PDF/A-2a, PDF/A-2b, PDF/A-2u
- PDF/A-3a, PDF/A-3b, PDF/A-3u

1.2.3 Conformance

- Standards:
  - ISO 32000-1 (PDF 1.7)
  - ISO 32000-2 (PDF 2.0)
  - ISO 19005-1 (PDF/A-1)
  - ISO 19005-2 (PDF/A-2)
  - ISO 19005-3 (PDF/A-3)
  - PAdES (ETSI EN 319 142) signature levels B-B, B-T, CMS
  - Legacy PAdES baseline signature (ETSI TS 103 172) B-Level and T-Level
  - Legacy PAdES (ETSI TS 102 778) Part 2 (PAdES Basic), Part 3 (PAdES-BES), and Part 4 (PAdES-LTV, Long Term Validation)
  - Long term signature profiles for PAdES (ISO 14533-3)
  - Cryptographic Suites (ETSI TS 119 312)
- Quality assurance: veraPDF test corpus and Isartor test suite

1.3 Service

The 3-Heights™ PDF to PDF/A Converter Service is a ready-to-use product that allows to install a Windows NT service process to automatically convert various types of images from watched folders into PDF files. The 3-Heights™ PDF to PDF/A Converter Service combines three programs in one executable.

1. A converting service, that can be run on Windows platforms. The service can be started, paused, stopped via the Windows service control panel and reports to the application log of the Windows event log panel.
2. A command line interface to control the Image to PDF Converter Service. By means of this interface the service can be installed, started, stopped and deleted.
3. A converter query program which can be used to retrieve information about available conversion options such as file type, compression, dithering, color depths, etc.

1.4 Operating Systems

The 3-Heights™ PDF to PDF/A Converter Service is available for the following operating systems:

- Windows Client 7+ | x86 and x64

‘+’ indicates the minimum supported version.

1.5 Digital Signatures

1.5.1 Overview

Digital signature is a large and slightly complex topic. This manual gives an introduction to digital signatures and describes how the 3-Heights™ PDF to PDF/A Converter Service is used to apply them. It does however not describe all the technical details.
1.5.2 Terminology

**Digital Signature** is a cryptographic technique of calculating a number (a digital signature) for a message. Creating a digital signature requires a private key from a certificate. Validating a digital signature and its authorship requires a public key. Digital Signature is a technical term.

**Electronic Signature** is a set of electronic data that is merged or linked to other electronic data in order to authenticate it. Electronic Signatures can be created by means of a digital signature or other techniques. Electronic Signature is a legal term.

<table>
<thead>
<tr>
<th>Abbreviations</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
</tr>
<tr>
<td>CMS</td>
</tr>
<tr>
<td>CRL</td>
</tr>
<tr>
<td>CSP</td>
</tr>
<tr>
<td>HSM</td>
</tr>
<tr>
<td>OCSP</td>
</tr>
<tr>
<td>PKCS</td>
</tr>
<tr>
<td>QES</td>
</tr>
<tr>
<td>TSA</td>
</tr>
<tr>
<td>TSP</td>
</tr>
</tbody>
</table>

1.5.3 Why Digitally Signing?

The idea of applying a digital signature in PDF is very similar to a handwritten signature: A person reads a document and signs it with its name. In addition to the name, the signature can contain further optional information, such as the date and location. A valid electronic signature is a section of data that can be used to:

- Ensure the integrity of the document
- Authenticate the signer of the document
- Prove existence of file prior to date (time-stamp)

Digitally signing a document requires a certificate and its private key. How to access and use a certificate is described in the chapter **Cryptographic Provider**.

In a PDF document, a digital signature consists of two parts:

**A PDF related part** This part consists of the PDF objects required to embed the signature into the PDF document. This part depends on the signature type (Document Signature, MDP Signature, see table below). Information such as name of the signer, reason, date, location is stored here. The signature may optionally have a visual appearance on a page of the PDF document, which can contain text, graphics and images.

This part of the signature is entirely created by the 3-Heights™ PDF to PDF/A Converter Service.

**A cryptographic part** A digital signature is based on a cryptographic checksum (hash value) calculated from the content of the document that is being signed. If the document is modified at a later time, the computed hash
value is no longer correct and the signature becomes invalid, i.e. the validation will fail and will report that the
document has been modified since the signature was applied. Only the owner of the certificate and its private
key is able to sign the document. However, anybody can verify the signature with the public key contained in
the certificate.

This part of the signature requires a cryptographic provider for some cryptographic data and algorithms.

The 3-Heights™ PDF to PDF/A Converter Service supports the following types of digital signatures:

- **Document Signature**  
  Check the integrity of the signed part of the document and authenticate the signer’s
  identity. One or more document signatures can be applied. A signed document can be modified and saved by
  incremental updates. The state of the document can be re-created as it existed at the time of signing.

- **MDP (Modification detection and prevention) Signature**  
  Enable detection of disallowed changes specified by
  the author. A document can contain only one MDP signature; which must be the first in the document. Other
  types of signatures may be present.

- **Document Time-stamp Signature**  
  A time-stamp signature provides evidence that the document existed at
  a specific time and protects the document’s integrity. One or more document time-stamp signatures can be
  applied. A signed document can be modified and saved by incremental updates.

### 1.5.4 What is an Electronic Signature?

There are different types of electronic signatures, which normally are defined by national laws, and therefore are
different for different countries. The type of electronic signatures required in a certain process is usually defined by
national laws. Quite advanced in this manner are German-speaking countries where such laws and an established
terminology exist. The English terminology is basically a translation from German.

Three types of electronic signatures are distinguished:

- Simple Electronic Signature “Einfache Elektronische Signatur”
- Advanced Electronic Signature “Fortgeschrittene Elektronische Signatur”
- Qualified Electronic Signature (QES) “Qualifizierte Elektronische Signatur”

All applied digital signatures conform to PDF/A and PAdES.

### Simple Electronic Signature

A simple electronic signature requires any certificate that can be used for digital signing. The easiest way to retrieve
a certificate, which meets that requirement, is to create a so called self-signed certificate. Self-signed means it is
signed by its owner, therefore the issuer of the certificate and the approver of the legitimacy of a document signed
by this certificate is the same person.

**Example:**

Anyone could create a self-signed certificate issued by “Peter Pan” and issued to “Peter Pan”. Using this certificate
one is able to sign in the name of “Peter Pan”.

If a PDF document is signed with a simple electronic signature and the document is changed after the signature
had been applied, the signature becomes invalid. However, the person who applied the changes, could at the
same time (maliciously) also remove the existing simple electronic signature and—after the changes—apply a new,
equally looking Simple Electronic Signature and falsify its date. As we can see, a simple electronic signature is neither
strong enough to ensure the integrity of the document nor to authenticate the signer.

This drawback can overcome using an advanced or Qualified Electronic Signature.
**Advanced Electronic Signature**

Requirements for advanced certificates and signatures vary depending on the country where they are issued and used.

An advanced electronic signature is based on an advanced certificate that is issued by a recognized certificate authority (CA) in this country, such as VeriSign, SwissSign, QuoVadis. In order to receive an advanced certificate, its owner must prove its identity, e.g. by physically visiting the CA and presenting its passport. The owner can be an individual or legal person or entity.

An advanced certificate contains the name of the owner, the name of the CA, its period of validity and other information.

The private key of the certificate is protected by a PIN, which is only known to its owner.

This brings the following advantages over a simple electronic signature:

- The signature authenticates the signer.
- The signature ensures the integrity of the signed content.

**Qualified Electronic Signature**

Requirements for qualified certificates and signatures vary depending on the country where they are issued and used.

A Qualified Electronic Signature is similar to an advanced electronic signature, but has higher requirements. The main differences are:

- It is based on a qualified certificate, which is provided as a hardware token (USB stick, smart card).
- For every signature it is required to enter the PIN code manually. This means that only one signature can be applied at a time.
- Certificate revocation information (OCSP/CRL) can be acquired from an online service. The response (valid, revoked, etc.) must be embedded in the signature.
- A time-stamp (TSP) that is acquired from a trusted time server (TSA) may be required.

This brings the following advantages over an advanced electronic signature:

- The signature ensures the certificate was valid at the time when the document was signed (due to the embedding of the OCSP/CRL response).
- The signature ensures the integrity of the time of signing (due to the embedding of the time-stamp).
- Legal processes that require a QES are supported.

**Note:** A time-stamp can be added to any type of signature. OCSP/CRL responses are also available for some advanced certificates.

### 1.5.5 How to Create Electronic Signatures

This is a simple example of how to create an electronic document signature. More detailed examples can be found in [How to Create Digital Signatures](#).

**Preparation Steps**

1. Identify whether an [Advanced Electronic Signature](#) or a [Qualified Electronic Signature](#) is required. For most automated processes an advanced signature is sufficient.
2. Identify regulatory requirements regarding the content and life cycle of the signature:
Is a time-stamp required to prove that the signature itself existed at a certain date and time?
Should validation information be embedded, in order to allow the signature to be validated long time after its generation?
Should the integrity of the validation material be protected?
Is a specific signature encoding required?

These requirements (or regulatory requirements) define the signature level that must be used.

3. Acquire a corresponding certificate from a CA.

For automated processes we recommend to use a HSM, an online signing service, or soft certificates. Other hardware such as USB tokens or Smart Cards are often cheaper, but limited to local interactive single-user applications.

When using an online signing service, ensure that it supports the required signature encoding.

4. Setup and configure the certificate’s Cryptographic Provider.

- In case the certificate resides on hardware such as an USB token or a Smart Card, the required middleware (driver) needs to be installed.
- In case the certificate is a soft certificate, it must be imported into the certificate store of a cryptographic provider.

5. Optional: Acquire access to a trusted time server (TSA) (preferably from the CA of your signing certificate).

6. Optional: Ensure your input documents conform to the PDF/A standard.

It is recommended to sign PDF/A documents only, because this ensures that the file’s visual appearance is well defined, such that it can be reproduced flawlessly and authentically in any environment. Furthermore, PDF/A conformance is typically required if the file is to be archived. Because signed files cannot be converted to PDF/A without breaking its signatures, files must be converted before signing.

Note: A detailed guidance on the use of standards for signature creation can be found in the technical report ETSI TR 119 100.

Application of the Signature

Apply the signature by providing the following information:

1. The Cryptographic Provider where the certificate is located
2. Values for the selection of the signing certificate (e.g. the name of the certificate)
3. Optional: Time-stamp service URL (e.g. “http://server.mydomain.com:80/tsa”)
4. Optional: Time-stamp service credentials (e.g. username:password)
5. Optional: Add validation information
6. Optional: Visual appearance of the signature on a page of the document (e.g. an image).

Example: Steps to Add an Electronic Document Signature

The 3-Heights™ PDF to PDF/A Converter Service applies PDF/A conforming signatures. This means if a PDF/A document is digitally signed, it retains PDF/A conformance.

In order to add an electronic document signature with the 3-Heights™ PDF to PDF/A Converter Service the following steps need to be done:

1. Provide the certificate name (Subject).
2. Apply settings for the signature, such as the reason text, or the visual appearance (color, position, etc).
3. Process the PDF document by a user which has access to the selected certificate and thereby add the signature.

The certificate name is provided with the switch `-cn`, the reason with the switch `-cr` and the provider (including the PIN to access the certificate's private key) with the switch `-cp`. A sample command looks like this:

```
-cn "Philip Renggli"
```
The visual appearance of the digital signature on a page of the resulting output-document looks as shown below:
2 Installation

2.1 Overview

The PDF to PDF/A Converter Service is configured by the file Pdf2PdfSvr.ini, which needs to be located in the same directory as the executable Pdf2PdfSvr.exe. Before starting the service, the configuration file needs to be adjusted. How this is done is described in the chapter Configuration File Pdf2PdfSvr.ini.

Once configured, the service can be created, started, paused, continued, stopped and deleted via the command line. To use the create and delete functions, administrator permissions are required. To start and stop the service, operator permissions are required.

When the service is running, it processes PDF documents that are copied or moved into watched folders. They are then renamed and moved to the folder Jobs. The renaming gives the PDF a 16 character long Time-stamp to create unique job tickets. This ensures there are no conflicts with documents that have the same name.

2.2 Windows

The 3-Heights™ PDF to PDF/A Converter Service comes as an MSI installer.

The installation of the software requires the following steps.

1. You need administrator rights to install this software.
2. Log in to your download account at http://www.pdf-tools.com. Select the product "PDF to PDF/A Converter Service". If you have no active downloads available or cannot log in, please contact pdfsales@pdf-tools.com for assistance.
   You will find different versions of the product available. We suggest to download the version, which is selected by default. A different version can be selected using the combo box.
   The product comes as an MSI (Microsoft Installer) package that provides an installation routine for installing and uninstalling the 3-Heights™ PDF to PDF/A Converter Service.
   The package installs the 64-bit version, which runs on 64-bit platforms only.
3. Start the MSI package and follow the steps in the installation routine.
4. Ensure the cache directory exists as described in chapter Special Directories.
5. Make sure your platform meets the requirements regarding color spaces and fonts described in chapters Color Spaces and Fonts respectively.
6. If you want to sign documents, proceed with setting up your cryptographic provider as described in chapter Cryptographic Provider.
7. (Optional) Download and install the 3-Heights™ OCR Enterprise Add-On and the OCR Engine as described in the respective manuals:
   - 3-Heights™ OCR Service: OcrService.pdf from the separate product kit.

2.3 Uninstall

If you have used the MSI for the installation, go to Start → 3-Heights™ PDF to PDF/A Converter Service → Uninstall...
2.4 **Note about the Evaluation License**

With the evaluation license the 3-Heights™ PDF to PDF/A Converter Service automatically adds a watermark to the output files.

2.5 **Special Directories**

2.5.1 **Directory for temporary files**

This directory for temporary files is used for data specific to one instance of a program. The data is not shared between different invocations and deleted after termination of the program.

The directory is determined as follows. The product checks for the existence of environment variables in the following order and uses the first path found:

**Windows**

1. The path specified by the `%TMP%` environment variable.
2. The path specified by the `%TEMP%` environment variable.
3. The path specified by the `%USERPROFILE%` environment variable.
4. The Windows directory.

2.5.2 **Cache Directory**

The cache directory is used for data that is persisted and shared between different invocations of a program. The actual caches are created in subdirectories. The content of this directory can safely be deleted to clean all caches. This directory should be writable by the application, otherwise caches cannot be created or updated and performance will degrade significantly.

**Windows**

- If the user has a profile:  
  `%LOCAL_APPDATA%\PDF Tools AG\Caches`
- If the user has no profile:  
  `<TempDirectory>\PDF Tools AG\Caches`

where `<TempDirectory>` refers to the **Directory for temporary files**.

2.5.3 **Font Directories**

The location of the font directories depends on the operating system. Font directories are traversed recursively in the order as specified below.

If two fonts with the same name are found, the latter one takes precedence, i.e. user fonts will always take precedence over system fonts.

**Windows**

1. `%SystemRoot%\Fonts`
2. User fonts listed in the registry key `\HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Fonts`. This includes user specific fonts from `C:\Users\<user>\AppData\Local\Microsoft\Windows\Fonts` and app specific fonts from `C:\Program Files\WindowsApps`
3. `directory Fonts`, which must be a direct sub-directory of where `Pdf2PdfSvr.exe` resides.
3 License Management

The 3-Heights™ PDF to PDF/A Converter Service requires a valid license in order to run correctly. If no license key is set or the license is not valid, then an error message will be printed to the service log.

3.1 License Features

The functionality of the 3-Heights™ PDF to PDF/A Converter Service contains one area to which the following license feature is assigned:

Signature  Signature creation.

The presence of this feature in a given license key can be checked in the Graphical License Manager Tool or by means of the Command Line License Manager Tool. The Interface Reference specifies in more detail which functions are included in this license feature.

3.2 License Installation and Management

There are two possibilities to pass the license key to the application:

1. The license key is installed using the GUI tool (graphical user interface). This is the easiest way if the licenses are managed manually. It is only available on Windows.
2. The license key is installed using the shell tool. This is the preferred solution for automated license management.

3.2.1 Graphical License Manager Tool

The GUI tool LicenseManager.exe is located in the bin directory of the product kit (Windows only).

List all installed license keys

The license manager always shows a list of all installed license keys in the left pane of the window. This includes licenses of other PDF Tools products. The user can choose between:

- Licenses available for all users. Administrator rights are needed for modifications.
- Licenses available for the current user only.
Add and delete license keys

License keys can be added or deleted with the “Add Key” and “Delete” buttons in the toolbar.

- The “Add key” button installs the license key into the currently selected list.

**Note:** Services run by default under the LOCAL SERVICE user, not under the current user.

- The “Delete” button deletes the currently selected license keys.

Display the properties of a license

If a license is selected in the license list, its properties are displayed in the right pane of the window.

3.2.2 Command Line License Manager Tool

The command line license manager tool licmgr is available in the bin\x86 and bin\x64 directory.

**Note:** The command line tool licmgr is not included in Windows platform kits, as the GUI tool is the recommended tool for managing licenses. A Windows licmgr shelltool is available in the Utilities & Tools section of your My PDF Tools customer account.

A complete description of all commands and options can be obtained by running the program without parameters:

```
licmgr
```

List all installed license keys

```
licmgr list
```

The currently active license for a specific product is marked with a * on the left side.

**Example:**

```
>licmgr list
Local machine:
   Product Name:
      1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
      1-YYYYY-YYYYY-YYYYY-YYYYY-YYYYY-YYYYY-YYYYY
      * 1-ZZZZZ-ZZZZZ-ZZZZZ-ZZZZZ-ZZZZZ-ZZZZZ-ZZZZZ
Current user:
```

Add and delete license keys

Install new license key:
licmgr store 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX

Delete old license key:

licmgr delete 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX

Both commands have the optional argument \texttt{-s} that defines the scope of the action:

\texttt{g} \quad \text{For all users}
\texttt{u} \quad \text{Current user}

\textbf{Display the properties of a license}

licmgr info 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX

Properties that invalidate the license are marked with an \texttt{X}, properties that require attention are marked with an \texttt{!}. In that case an additional line with a comment is displayed.

\textbf{Example:}

\begin{verbatim}
>licmgr info 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
- Product: Product Name
- Features: Feature1,Feature2
- Intended use: Development
- Watermark: No
- Platform: Windows
- Installation: Yes
- Activation: 2018-05-07
   (The license has not yet been activated.)
- Expiration: Does not expire
- Maintenance: 2019-04-27
\end{verbatim}

\section{3.3 License Selection and Precedence}

\subsection{3.3.1 Selection}

If multiple keys for the same product are installed in the same scope, only one of them can be active at the same time.

Installed keys that are not selected are not considered by the software!

\textbf{In the Graphical User Interface} use the check box on the left side of the license key to mark a license as selected.
With the Command Line Interface use the select subcommand:

```
licmgr select 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

### 3.3.2 Precedence

License keys are considered in the following order:

1. License selected for the current user
2. License selected for the current user (legacy key format)
3. License selected for all users
4. License selected for all users (legacy key format)

The first matching license is used, regardless whether it is valid or not.

### 3.4 Key Update

If a license property like the maintenance expiration date changes, the key can be update directly in the license manager.

**In the Graphical User Interface** select the license and press the button “Update Key” in the toolbar:

**With the Command Line Interface** use the update subcommand:

```
licmgr update 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

### 3.5 License activation

New licenses keys have to be activated (except for OEM licenses).

**Note:** Licenses that need activation have to be installed in the license manager and must not be passed to the component at runtime.

The license activation is tied to a specific computer. If the license is installed at user scope, the activation is also tied to that specific user. The same license key can be activated multiple times, if the license quantity is larger than 1.

Every license key includes a date, after which the license has to be activated, which is typically 10 days after the issuing date of the key. Prior to this date, the key can be used without activation and without any restrictions.

#### 3.5.1 Activation

The License can be activated directly within the license manager. Every activation increases the activation count of the license by 1.
It is recommended to add a comment to the activation request which helps keeping track of all activations for a specific license key. In case of problems it also helps us providing support.

The comment is stored in the activation database as long as the license key remains activated. Upon deactivation it is deleted from the database immediately.

All activations and the corresponding comments can be examined using the Load online properties function of the license manager. The information is accessible to anyone with access to the license key.

**In the Graphical User Interface** select the license and press the button “Activate license” in the toolbar:

![Image](image.png)

It is recommended to add a comment to the activation request by using the subsequent dialog box.

**With the Command Line Interface** use the activate subcommand:

```
licmgr activate 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

Note that the key has to be installed first.

It is recommended to add a comment to the activation request by using the -c or -cd option:

```
licmgr activate -cd 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
licmgr activate -c "custom comment" 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

### 3.5.2 Reactivation

The activation is tied to specific properties of the computer like the MAC address or host name. If one of these properties changes, the activation becomes invalid and the license has to be reactivated. A reactivation does not increase the activation count on the license.

The process for reactivation is the same as for the activation.

**In the Graphical User Interface** the button “Activate license” changes to “Reactivate license”:

![Image](image.png)

**With the Command Line Interface** the subcommand activate is used again:

```
licmgr activate 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

### 3.5.3 Deactivation

To move a license to a different computer, it has to be deactivated first. Deactivation decreases the activation count of the license by 1.
The process for deactivation is similar to the activation process.

**In the Graphical User Interface** select the license and press the button “Deactivate license” in the toolbar:

![Image of Graphical User Interface](image)

**With the Command Line Interface** use the deactivate subcommand:

```
licmgr deactivate 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

### 3.6 Proxy Setting

A proxy URL can be configured for computers that cannot access the internet without a web proxy.

**Note:** The proxy must allow connections via HTTP CONNECT to the server www.pdf-tools.com:443.

**In the Graphical User Interface** press the button “Settings” in the toolbar:

![Image of Graphical User Interface](image)

and enter the proxy URL in the respective field:

![Image of Settings window](image)

### 3.7 Offline Usage

The following actions in the license manager need access to the internet:

- **License Activation**
- **License Reactivation**
- **License Deactivation**
- **Key Update**

On systems without internet access, a three step process can be used instead, using a form on the PDF Tools website.

#### 3.7.1 First Step: Create a Request File

**In the Graphical User Interface** select the license and use the dropdown menu on the right side of the button in the toolbar:
3.7 With the Command Line Interface

use the `-fs` option to specify the destination path of the request file:

```
licmgr activate -fs activation_request.bin 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

License Deactivation: When saving the deactivation request file, the license is **deactivated immediately** and cannot be used any further. It can however only be activated again after completing the deactivation on the website.

3.7.2 Second Step: Use Form on Website


Upload the request by dragging it onto the marked area:

Upon success, the response will be downloaded automatically if necessary.

3.7.3 Third Step: Apply the Response File

In the Graphical User Interface select the license and use the dropdown menu on right side of the button in the toolbar:

With the Command Line Interface use the `-fl` option to specify the source path of the response file:

```
licmgr activate -fl activation_response.bin 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```
3.8 License Key Versions

As of 2018 all new keys will have the format 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX. Legacy keys with the old format 0-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX are still accepted for a limited time period.

For compatibility reasons, old and new version keys can be installed side by side and one key of each version can be selected at the same time. In that case, the software always uses the new version.

3.9 License Key Storage

Depending on the platform the license management system uses different stores for the license keys.

3.9.1 Windows

The license keys are stored in the registry:

- “HKLM\Software\PDF Tools AG” (for all users)
- “HKCU\Software\PDF Tools AG” (for the current user)

3.10 Troubleshooting

3.10.1 License key cannot be installed

The license key cannot be installed in the license manager application. The error message is: "Invalid license format."

Possible causes:

- The license manager application is an older version that only supports the legacy key format.

Solution

Use a current version of the license manager application or use a license key in the legacy key format if available.

3.10.2 License is not visible in license manager

The license key was successfully installed previously but is not visible in the license manager anymore. The software is still working correctly.

Possible causes:

- The license manager application is an older version that only supports the legacy key format.

Solution

Use a current version of the license manager application.
3.10.3 License is not found at runtime

The license is not found at runtime by the software. The error message is: "No license key was set."

Possible causes:

- The license key is actually missing (not installed).
- The license key is installed but not selected in the license manager.
- The application is an older version that only supports the legacy key format, while the license key has the new license format.
- The software was not restarted after registering the license.

Solution
Install and select a valid license key that is compatible with the installed version of the software or use a newer version of the software. The new license key format is supported starting with version 4.10.26.1
For compatibility reasons, one license key of each format can be selected at the same time.

3.10.4 Eval watermark is displayed where it should not

The software prints an evaluation watermark onto the output document, even if the installed license is a productive one.

Possible causes:

- There is an evaluation license key selected for the current user, that takes precedence over the key for all users.

Note: The software might be run under a different user than the license manager application.

- An evaluation license key that is passed at runtime takes precedence over those selected in the license manager.
- There is an evaluation license key selected with a newer license format that takes precedence over the key in the older format.
- The software was not restarted after changing the license key from an evaluation key to a productive one.

Solution
Disable or remove all evaluation license in all scopes, check that no evaluation key is passed at runtime and restart the software.

3.10.5 Activation is not recognized

The license is installed and activated in the license manager, but the software does not recognize it as activated. The error message is: "The license has not been activated."

Possible causes:

- There is an unregistered license key selected for the current user, that takes precedence over the key for all users. This leads to an error even if the same license is registered for all users.
**Note:** The software might be run under a different user than the license manager application.

- A license key that is passed at runtime takes precedence over those selected in the license manager. This leads to an error even if the same license is registered in the license manager.

**Note:** Licenses that need activation have to be installed in the license manager and must not be passed to the component at runtime.

- The software was not restarted after activating the license.

**Solution**
Disable, remove or activate all unregistered licenses in all scopes, check that no key is passed at runtime and restart the software.

### 3.10.6 Activation is invalidated too often

The license activation is invalidated regularly, for no obvious reason.

**Possible causes:**
- One of the properties used to calculate the system fingerprint is changing frequently.

**Solution**
Update to a newer version of the PDF Tools product, deactivate the license key using the new license manager and activate it again. After that, an improved fingerprinting algorithm is used. Deactivation and activation have to be **executed separately**, a reactivation of the license in one step does not change the fingerprinting algorithm and thus does not solve the problem.

**Note:** After this procedure, older products might not recognize the activation as valid anymore. Reactivating the license using an old license manager will revert the activation to the old fingerprinting algorithm.

### 3.10.7 Connection to the licensing service fails

The license activation/deactivation/update fails because the license manager cannot reach the licensing server.
The error message depends on the platform and the exact error condition.

**Possible causes:**
- The computer is not connected to the internet.
- The connection is blocked by a corporate firewall.

**Solution**
Make sure that the computer is connected to the internet and that the host `www.pdf-tools.com` is reachable on port 443 (HTTPS).
If this is not possible, try [Offline Usage](#) instead.

### 3.10.8 Offline usage fails due to a request/response mismatch

The offline license activation/deactivation/update fails because the response file does not match the request file. The error message is: "Mismatch between request and response."

**Possible causes:**

- The response file is applied to a different machine than the request file was created.
- The response file as applied to a different user than the request file was created.
- The response file was applied to a specific user while the request was created for all users, or vice versa.
- The response file is applied to the wrong license key.
- Another request file has been created between creating the request file and applying the response file.
- The license key was updated between creating the request file and applying the response file.
- The license key was removed and re-added between creating the request file and applying the response file.

**Solution**

Delete any old request and response files to make sure they are not used by accident.

Retry the entire process as outlined in [chapter 3.7](#) and refrain from making any other license-related actions between creating the request file and applying the response file.

Make sure that the response file is applied to exactly the same license key in exactly the same location (machine, all users or specific user) where the request file was created.
4 Getting Started

4.1 Configuration

Before starting the PDF to PDF/A Converter Service for the first time, the file `Pdf2PdfSvr.ini` needs to be modified. Editing this file while the PDF to PDF/A Converter Service is running has no impact. The service first needs to be stopped and restarted after the modification. When opening this file with a text editor, it looks like this:

```ini
[Pdf2PdfSvr]
AutoDelete=True
Threads=1
Thread1=-w C:\Pdf2PdfSvr
-oi "C:\Winnt\system32\spool\drivers\color\sRGB Color Space Profile.icm"
-cs "C:\Winnt\system32\spool\drivers\color\USWebCoatedSWOP.icc"
-cn "Philip Renggli" -cr "I am the author of the document"
```

The meaning of these keys and values in this example is as follows:

- **AutoDelete=True**    This option automatically deletes a PDF file after it is processed successfully. When set to False, the processed file will be copied to the sub directory Succeeded.

- **Threads**= The given value stands for the total number of concurrent threads. Each thread can have its own assigned settings. One thread corresponds to one watched folder.

- **Thread1**= Sets the options such as name of watched folder and settings etc. for thread 1.

  - **-w C:\Pdf2PdfSvr** (required) Creates a watched folder with the given name for this thread. The path must be an absolute path. Network mapped drive letters or relative paths or driver letters mapped via the subst command are not recognized, because the service process per default runs under the “LocalSystem” account. (The user can be changed as described in chapter Managing the Service.)

  - **-oi "C:\Winnt\...\sRGB Color Space Profile.icm"** (optional) Sets the color profile for the output intent. An output intent is required for a PDF/A compliant document. If the document already contains an output intent, this option does nothing.

  - **-cs "C:\Winnt\...\USWebCoatedSWOP.icc"** (optional) Sets the color profile for color spaces of embedded images.

  - **-cn "Philip Renggli"** (optional) Sets the signature name.

  - **-cr "I am the author of the document."** (optional) Sets a signature reason.

This means that any PDF document that is moved or copied to the folder `C:\Pdf2PdfSvr` will be processed by the service and converted to a PDF/A document. In this configuration example the PDF/A document will have an output intent profile, a color space profile and will be digitally signed.

**Note:** Any string, such as a file name, that contains spaces must be enclosed in quotation marks. E.g. if the watched folder contains spaces in its path, the entire path needs to be quoted: `-w "C:\A path\with spaces"`. 
4.1.1 Retrieve Information about Available Options and Settings

A quick overview over all configuration options and service control commands that the 3-Heights™ PDF to PDF/A Converter Service supports can be output in the form of a usage message on the command line.

To display this information, first open a Windows command line (cmd.exe) and then type:

```
Pdf2PdfSvr
```

(See also Service Control Commands.)

A short overview over all the options that can be configured in the `Pdf2PdfSvr.ini` is displayed when typing the following in a Windows command line:

```
Pdf2PdfSvr -i
```

(See also General Settings, Color Profiles, Digital Signatures, and OCR.)

4.2 Managing the Service

Once the configuration is done, the service can be started and controlled by executing `Pdf2PdfSvr.exe` on the command line. The path can be omitted if the `Pdf2PdfSvr.exe` is included in the `%PATH%` environment variable.

**Note:** It is essential that the executable `Pdf2PdfSvr.exe` and the configuration file `Pdf2PdfSvr.ini` be on a non-mapped drive.

**Note:** To create or delete the service, administrator permissions are required.

1. To create the service, use the option `-c`.

```
Pdf2PdfSvr -c
```

After executing this command, the service is created. It is now visible in the “Computer Management” window under “Services”. To open the “Computer Management” window, go to Start → Control Panel → Administrative Tools → Computer Management or simply right-click the icon “My Computer” on the desktop and select “manage”. If the services was created correctly it appears as “3-Heights™ PDF to PDF/A Converter Service” as shown in the image below.
2. By default, the 3-Heights™ PDF to PDF/A Converter Service runs in the “LocalSystem” account. After the service has been created, the user can be changed. This will be required in a situation where a network share is used as a watched folder and the process needs to run under a user with the appropriate access permission rights, since the account “LocalSystem” does not have any permissions on remote systems.

If digital signatures are to be applied, the certificates might be bound to a user or even an interactive session. Check your Cryptographic Provider for requirements regarding the user.

To change the user, right-click the service in the Services window and select “Properties”. Then change the user in the tab “Log On”.

3. After its creation, the service can be started with the option `-s`.

```
Pdf2PdfSvr -s
```

4. Now the 3-Heights™ PDF to PDF/A Converter Service is up and running, and files can be moved, copied or drag-and-dropped into the watched folder.

5. To stop the service, use the option `-t`.

```
Pdf2PdfSvr -t
```

To restart use `-s` again.

6. To delete the service use the option `-d`.

```
Pdf2PdfSvr -d
```

### 4.2.1 State Diagram of the Service

The 3-Heights™ PDF to PDF/A Converter Service behaves as described in the state diagram below:

If “Stop” is called when the service is in the state “Paused”, the current job is aborted. This means the current page is finished processing, then the job is terminated.

If “Stop” is called when the service is the state “Running”, the current job (all pages) is finished. Then the service is stopped.
4.3 Using the Service

Once the service is created and started, the watched folders configured in Pdf2PdfSvr.ini are created automatically. In each watched folder, the following sub-folders are created:

- Jobs
- InProgress
- Succeeded
- Failed
- PDFs
- Logs

When a file is moved, copied, or drag-and-dropped into the configured watched folder, the service will do the following:

1. Each file is moved to the sub folder Jobs. While moving, the file is renamed by adding a 16 character long job-number prefix. This ensures a well defined processing order and unique file names.
2. A worker-thread takes the file from the folder Jobs and moves it to InProgress. The file is then processed.
3. Depending on the outcome of the processing, the following is done:

   **The file was processed successfully**
   - The input file is moved to the folder Succeeded or it is deleted, depending on whether AutoDelete or AutoDeleteAll is set to true or false in the configuration file Pdf2PdfSvr.ini.
   - The converted document is stored in PDFs.

   **The file was not processed successfully**
The input file is moved to the folder Failed or it is deleted, depending on whether AutoDeleteALL is set to true or false in the configuration file Pdf2PdfSvr.ini.

A log file is created in the folder Logs. The file name is the same as the input file with the extension substituted by .txt.

4. In any case, an entry in the log file of this thread is created.

4.4 Log Files

There are two types of log files.

The log file per thread Each thread (watched folder) has a log file. The log file resides in the same directory as the executable Pdf2PdfSvr.exe and the configuration file Pdf2PdfSvr.ini. It is named Pdf2PdfSvr-log-‹n›.txt, where the number of the log file ‹n› is increased whenever the service is re-started. The log file is locked by the service as long as the service is running.

- The log file contains general messages (including a time stamp that is not shown here) such as:

  - [1] Worker thread for directory C:\Pdf2PdfSvr\Folder started.

- Error messages such as:

  * Error 0 while opening file C:\Pdf2PdfSvr\Folder\InProgress\Job-...

The error log file per document For each document which failed to be processed successfully, there is an error log file created. The log file is written to the sub-folder Logs and has the same file name as the input document, with extension .txt. For more details on the possible log messages in the per-document log file see Log File. There is no error log available if a document cannot be opened (e.g. it is not an appropriate document type).
5 User’s Guide

5.1 Process Description

The workflow of the PDF to PDF/A Conversion is outlined in the graphic below.

1. **License Check**: The license is checked.
2. **Pre-Analysis**: The input document is analyzed. If the document already conforms to the requested standard it
is copied.
If the required PDF/A level (e.g. level U or A) cannot be met, the conversion is aborted with an error¹.
If the target standard is PDF/A-1 and a file contains transparency or other elements that cannot be converted to
PDF/A-1, the target standard is upgraded to PDF/A-2 if the option \texttt{-ad} is set.
If the input document contains non-convertible elements the conversion is stopped, except if convert-always is
enabled.
3. **Conversion**: The actual conversion is performed.
The conversion is stopped, e.g. if an OCR error occurs, a required font is not found in the installed font directories,
 or linearization fails. In this case, no meaningful output document is created.
If actions had to be taken that might have altered the visual appearance of the file or crucial data had to be
removed, a conversion error is generated (see chapter Conversion Errors below).
4. **Post-Analysis**: Finally, the resulting PDF document is validated. If the resulting document does not meet the
requested standard a post-analysis error is raised.

5.1.1 Conversion Steps

The goal of the conversion is to create a document which is conforming to the PDF/A ISO standard.
If the analysis of the document indicates a conversion to the requested standard is possible, the following steps are
performed:

- Embed and subset non-embedded font programs
- Replace device specific color spaces with CIE-based color spaces
- Add a GTS_PDFA output intent
- Remove prohibited entries
- Remove entries with a default value
- Remove entries with unknown values
- Add mandatory entries
- Add XMP metadata if missing or fix inconsistent XMP metadata
- Apply implicit optimization functions (e.g. replace and subset embedded fonts)
- Apply implicit repair functions (to conform with ISO19005-1 chapter 6.1)

If the analysis indicates a conversion is not possible, a “best effort” conversion can be forced. In this case the output
may or may not be PDF/A conformant. Use the post analysis feature in order to detect, whether or not the out-
put is conformant. It is also possible that the output file looks visually different to the input file due to the forced
conversion.

5.1.2 Conversion Errors

The conversion error indicates that during conversion, actions had to be taken that might have altered the visual
appearance of the file or crucial data had to be removed.

\textbf{Note:} The resulting document conforms to PDF/A nonetheless.

The following issues may result in a conversion error:

- Optional content (layers) removed (PDF/A-1 only)
- Prohibited annotation type converted to stamp
- Prohibited action removed
- Embedded files removed
- Transparency removed (PDF/A-1 only)

¹ Automatic downgrades can be deactivated using the option \texttt{-ad}.
Character from show string removed because glyph missing in font
Unconvertible metadata

All conversion events are written to the log file. The description and location allow to identify potential problems quickly.

For a complete list of conversion events that can lead to a conversion error see the option \texttt{-cem}.

**Handling Conversion Errors**

We suggest checking, which conversion errors are tolerable in your process and which must be considered critical. Set the option \texttt{-cem} to include critical errors only.

Conversion errors can often be resolved by optimizing the 3-Heights™ PDF to PDF/A Converter Service’s options and or installation:

1. Conversion errors can be minimized by converting to PDF/A-2 instead of PDF/A-1. PDF/A-2 allows some features of newer versions of the PDF Reference, e.g. transparency, optional content (layers), or embedded files.
2. If fonts were substituted, the missing fonts should be installed (see Fonts).
3. For documents with non convertible XMP Metadata it is recommended to update the PDF creating software to generate valid XMP metadata.
4. How signed documents can be converted to PDF/A is explained in the following blog post by Dr. Hans Bärfuss.

In case of a conversion error, the output file is best presented to a user to decide whether or not the conversion result is acceptable. The conversion log file is helpful to display a meaningful message, e.g. “Embedded files have been removed during PDF/A conversion.” Also, all conversion events are written to the log file, indicating the cause of the conversion error.

For fully automated processes or documents which the user cannot accept, a fallback conversion can be added. For different conversion errors, different fallback conversions might be required:

1. If embedded files were removed, they can be extracted using the product 3-Heights™ PDF Extract and then converted to PDF/A by the 3-Heights™ PDF Document Converter.
2. Other conversion errors can be dealt with by creating an image based PDF using the 3-Heights™ PDF to Image Converter, which renders all pages and replaces their content with the resulting image. Optionally, some data such as bookmarks, links, or document metadata can be preserved.

5.1.3 Post Analysis

The post analysis step checks, whether or not the output file conforms to the requested standard. A post analysis error indicates that the output file is not PDF/A.

In case of a post analysis error, the conversion can be repeated with the option \texttt{-rd}. The log file then indicates why the post analysis failed. Often the issue can be resolved, e.g. by installing missing fonts (see Fonts) or with the option \texttt{-ad}.

If the file cannot be converted, a meaningful fallback could be the conversion to an image based PDF as described above in chapter Handling Conversion Errors.

5.2 What is PDF/A?

5.2.1 PDF/A-1

The PDF/A-1 format is described in the international standard ISO-19005-1. It is based on the PDF 1.4 reference and has some additional requirements. It is beneficial to have a general understanding of PDF/A. Here is a brief overview of how to create a PDF/A document from a non-PDF/A document.

1. A PDF/A has requirements about meta data and the structure of the file. The PDF to PDF/A Converter takes care of this and the user does not have to apply any settings. However he can provide the XMP meta data himself if desired.
2. In PDF/A, colors (including grayscale and black/white) must not be represented in a device color space (DeviceRGB, DeviceCMYK, DeviceGray). Suitable default color space profiles to substitute the device color spaces, one for RGB, CMYK and grayscale respectively can be provided by the user. In addition, or alternatively, one color space profile can be embedded as output intent. In this latter case, device colors are automatically managed by the output intent if the color can be represented in the space given by the color space profile in the output intent.
   
   If the converter encounters unmanaged colors, e.g. because no color space profile was set, then a calibrated color space is generated automatically, one RGB and one grayscale, for RGB and grayscale colors respectively. If unmanaged CMYK colors are encountered, a default CMYK output intent is embedded.
3. Fonts used in visible text must be embedded. This is automatically done by the Converter.
4. For PDF/A-1a: The original document structure information will be retained when converting the file to PDF/A. However, new tags will not be added and the structure will not be changed. To create a PDF/A-1a conforming file, the original file must have been created with the required structure and tagging. Otherwise, a PDF/A-1b file will be produced.

5.2.2 What is the difference between PDF/A-1b and PDF/A-1a?

PDF/A-1a has additional specifications on top of PDF/A-1b. These are:
1. The encoding of fonts must meet additional requirements, e.g. include a ToUnicode mapping (ISO 19005-1, chapter 6.3.8)
2. The document must contain a logical structure (ISO 19005-1, chapter 6.8)

The idea of the PDF/A-1a requirements is mainly to provide support for disabled people, i.e. by providing the required information needed for applications that support the read out loud feature.

The logical structure of the document is a description of the content of the pages. This description has to be provided by the creator of the document. It consists of a fine granular hierarchical tagging that distinguishes between the actual content and artifacts (such as page numbers, footers, layout artifacts, etc.). The tagging provides a meaningful description. Examples are “This is a Header”, “This color image shows a small sailing boat at sunset”, etc. One can easily understand this information cannot be generated automatically, it needs to be provided. This is one of the reasons why not every PDF document can be converted to PDF/A-1a.

5.2.3 PDF/A-2

PDF/A-2 is described in ISO 19005-2. It is based on ISO 32000-1, the standard for PDF 1.7. PDF/A-2 is meant as an extension to PDF/A-1. The second part shall complement the first part and not replace it. The most important differences between PDF/A-1 and PDF/A-2 are:

- The list of compression types has been extended by JPEG2000
- Transparent contents produced by graphic programs are allowed
- Optional contents (also known as layers) can be made visible or invisible
- Multiple PDF/A files can be bundled in one file (collection, package)
- The additional conformity level U (Unicode) allows for creating searchable files without having to fulfill the strict requirements of the conformity level A (accessibility)
- File size can be reduced using compressed object and XRef streams
Documents that contain features described above, in particular layers or transparency, should therefore be converted to PDF/A-2 rather than PDF/A-1.

5.2.4 PDF/A-3

PDF/A-3 is described in ISO 19005-3. It is based on ISO 32000-1, the standard for PDF 1.7. PDF/A-3 is an extension to PDF/A-2. The third part shall complement the second part and not replace it. The only two differences between PDF/A-2 and PDF/A-3 are:

- Files of any format and conformance may be embedded. Embedded files need not be suitable for long-term archiving.
- Embed files can be associated with any part of the PDF/A-3 file.

5.3 Color Spaces

5.3.1 Colors in PDF

The PDF format supports a range of color spaces:

**Device Color Spaces (DeviceGray, DeviceRGB, and DeviceCMYK)** These are also referred to as uncalibrated color spaces, because they cannot be used to specify color values such that colors are reproducible in a predictable way on multiple output devices.

**CIE-based Color Spaces (CalGray, CalRGB, Lab, ICCBased)** These are also referred to as device-independent color spaces, because they are inherently capable of specifying colors which can be reliably reproduced on multiple output devices.

**Special Color Spaces (Separation and DeviceN)** These require an alternate color space from one of the previous two groups to allow the PDF consumer to simulate the color on devices which do not support the special color space.

Colors can occur in the following objects of a PDF/A document:

- Raster images (also inline images)
- Text and Vector objects such as lines and curves
- Annotations
- Shading patterns
- Transparency blending (PDF/A-2 and later)

**ICC Color Profiles**

An ICC (International Color Consortium) profile is a file format which can be used to describe the color characteristics of a particular device. For example for the correct color reproduction when an image from a scanner or camera is displayed on a device, such as a monitor or printer. Color profiles are usually provided with the operating system (OS), on a Windows System, they can be found at the following location:

%SystemRoot%\system32\spool\drivers\color

Alternatively, additional profiles can be found here:

- [http://www.color.org/srgbprofiles.html](http://www.color.org/srgbprofiles.html)
Please note that most color profiles are copyrighted, therefore you should read the license agreements on the above links before using the color profiles. The PDF to PDF/A Converter will try to locate color profiles automatically in the %SystemRoot%\system32\spool\drivers\ color folder as needed.

**PDF/A Requirements**

In PDF/A the usage of uncalibrated color spaces (DeviceGray, DeviceRGB, and DeviceCMYK) is prohibited because colors that are specified in this way cannot be reproduced reliably on multiple output devices. Therefore, when converting to PDF/A, all device color spaces should be replaced by CIE-based color spaces. There is one exception to this rule: An uncalibrated color is tolerated if the output intent holds an ICC color profile with which this color can be represented. (E.g. a grayscale color can be represented in an RGB color profile, but a CMYK color cannot.)

The 3-Heights™ PDF to PDF/A Converter Service uses the following strategy:

- For each device color space (DeviceGray, DeviceRGB, and DeviceCMYK) an ICC color profile can be specified to be used as substitute for the respective device color space.
- One ICC color profile can be set to be used in the output intent.
- During conversion, if a device color space is encountered then the following is done:
  - If an output intent was set that is capable of managing this color, no action is needed.
  - Otherwise, if an ICC color profile is set to substitute this device color space then this color profile is used.
  - Otherwise, for DeviceRGB and DeviceGray color spaces: A calibrated color space (CalRGB\(^2\) and CalGray respectively) is generated and used as a substitute.
  - Otherwise, for DeviceCMYK color spaces:
    - If the output intent is not set, then a default CMYK ICC color profile is used for the output intent.
    - If the output intent holds a non-CMYK ICC color profile, then a default CMYK ICC color profile is generated and used as a substitute for DeviceCMYK.

The above strategy is motivated by the fact that CalRGB and CalGray color spaces occupy very little memory in comparison to ICC color profiles. Also note that the primary purpose of the output intent in a PDF document is to describe the characteristics of the device on which a document is intended to be rendered. Traditionally, the target device is a printer, which motivates CMYK output intents. The default CMYK color profile USWebCoatedSWOP.icc is provided in the sub-directory bin\icc.

### 5.4 Fonts

The PDF/A standard requires all fonts to be embedded in the PDF file. This ensures that the future rendering of the textual content of a conforming file matches, on a glyph by glyph basis, the appearance of the file as originally created.

Hence, if non-embedded fonts in a PDF are used, the font must be embedded. For this, a matching font has to be found in the Font Directories. The the option `-fd` should be used to define additional directories. The default font directories are listed in the chapter Font Directories.

It is important that the Font Directories contain all fonts that are used for the input files.

Fonts should be added to one of the Font Directories, if the post analysis returns validation errors like the following:

```
"output.pdf", 9, 20, 0x00418704, "The font ShinGo must be embedded.", 1
```

#### 5.4.1 Font Cache

A cache of all fonts in all Font Directories is created. If fonts are added or removed from the font directories, the cache is updated automatically.

\(^{2}\) The generated CalRGB color space is an approximation to the ICC color profile sRGB Color Space Profile.icm.
In order to achieve optimal performance, make sure that the cache directory is writable for the 3-Heights™ PDF to PDF/A Converter Service. Otherwise the font cache cannot be updated and the font directories have to be scanned on each program startup.

The font cache is created in the subdirectory `<CacheDirectory>/Installed Fonts` of the Cache Directory.

### 5.4.2 Font Configuration File `fonts.ini`

The font configuration file is optional. It can be used to control the embedding of fonts.

The file `fonts.ini` must reside at the following location:

**Windows:** In a directory named Fonts, which must be a direct sub-directory of where `Pdf2PdfSvr.exe` resides.

`fonts.ini` uses the INI file format and has two sections. The section `[fonts]` is ignored by the 3-Heights™ PDF to PDF/A Converter Service, so you may remove it. In the section `[replace]` font replacement rules of the form `key=value` can be defined. The key specifies the font that is to be replaced. The key should match the name of the font mentioned in the pre-analysis of the 3-Heights™ PDF to PDF/A Converter Service, e.g. "ShingGo" for:

```
"file.pdf", 9, 20, 0x00418704, "The font ShinGo must be embedded.", 1
```

The value should match the true type name of an installed font. Do not replace any standard fonts (Helvetica, Arial, Times, TimesNewRoman, Courier, CourierNew, Symbol, and ZapfDingbats).

Please note that this feature should be used with care. Replacing a font with another might change the visual appearance of the file because of different glyph shapes, metrics or glyphs that are not available in the replacement font. Embedding another font might also have legal implications.

**Example:** Replace MS-Mincyo with MS-Mincho

```
[replace]
MS-Mincyo=MS-Mincho
```

This rule defines, that in order to embed a font program for font MS-Mincyo the font MS-Mincho should be used. This rule is useful, because both names are possible transliterations of the same Japanese font. However, the official transliteration used by the actual font is MS-Mincho.

### 5.5 Cryptographic Provider

In order to use the 3-Heights™ PDF to PDF/A Converter Service’s cryptographic functions such as creating digital signatures, a cryptographic provider is required. The cryptographic provider manages certificates, their private keys and implements cryptographic algorithms.

The 3-Heights™ PDF to PDF/A Converter Service can use various different cryptographic providers. The following list shows, for which type of signing certificate which provider can be used.

**USB Token or Smart Card** These devices typically offer a PKCS#11 interface, which is the recommended way to use the certificate → [PKCS#11 Provider](#).

On Windows, the certificate is usually also available in the [Windows Cryptographic Provider](#).

Note that in any case, signing documents is only possible in an interactive user session.

**Hardware Security Module (HSM)** HSMs always offer very good PKCS#11 support → [PKCS#11 Provider](#).

For more information and installation instructions consult the separate document [TechNotePKCS11.pdf](#).
Soft Certificate  Soft certificates are typically PKCS#12 files that have the extension .pfx or .p12 and contain the signing certificate as well as the private key and trust chain (issuer certificates). Soft certificate files cannot be used directly. Instead, they must be imported into the certificate store of a cryptographic provider.

- **All Platforms**: The recommended way of using soft certificates is to import them into a store that offers a PKCS#11 interface and use the **PKCS#11 Provider**. For example:
  - A HSM
  - openCryptoki on Linux
  For more information and installation instructions of the above stores consult the separate document [TechNotePKCS11.pdf](#).
- **Windows**: If no PKCS#11 provider is available, soft certificates can be imported into Windows certificate store, which can then be used as cryptographic provider → [Windows Cryptographic Provider](#).

Signature Service  Signature services are a convenient alternative to storing certificates and key material locally. The 3-Heights™ PDF to PDF/A Converter Service can use various different services whose configuration is explained in the following sections of this documentation:

- SwissSign Digital Signing Service
- Swisscom All-in Signing Service
- GlobalSign Digital Signing Service
- QuoVadis sealsign

### 5.5.1 PKCS#11 Provider

PKCS#11 is a standard interface offered by most cryptographic devices such as HSMs, USB Tokens or sometimes even soft stores (e.g. openCryptoki).

More information on and installation instructions of the PKCS#11 provider of various cryptographic devices can be found in the separate document [TechNotePKCS11.pdf](#).

**Configuration**

**Provider**  Option `-cp`

The provider configuration string has the following syntax:

```
"<PathToDll>;<SlotId>;<Pin>"
```

**<PathToDll>** is the path to driver library filename, which is provided by the manufacturer of the HSM, UBS token or smart card. Examples:

- The CardOS API from Atos (Siemens) uses siecap11.dll
- The IBM 4758 cryptographic coprocessor uses cryptoki.dll
- Devices from Aladdin Ltd. use etpkcs11.dll
- The SuisseID USB Tokens use cvP11.dll

Please note that the sale of SuisseID will be discontinued as of 31. December 2019. On 15. December 2021 the SuisseID certificates will be revoked for regulatory reasons.

**<SlotId>** is optional, if it is not defined, it is searched for the first slot that contains a running token.

**<Pin>** is optional, if it is not defined, the submission for the pin is activated via the pad of the token.

If this is not supported by the token, the following error message is raised when signing: “Private key not available.”

Example:
Interoperability Support

The following cryptographic token interface (PKCS#11) products have been successfully tested:

- SafeNet Protect Server
- SafeNet Luna
- SafeNet Authentication Client
- IBM OpenCryptoki
- CryptoVision
- Siemens CardOS
- Utimaco SafeGuard CryptoServer

Selecting a Certificate for Signing

The 3-Heights™ PDF to PDF/A Converter Service offers different ways to select a certificate. The product tries the first of the following selection strategies, for which the required values have been specified by the user.

1. **Certificate fingerprint**
   Option `-cfp`
   SHA1 fingerprint of the certificate. The fingerprint is 20 bytes long and can be specified in hexadecimal string representation, e.g. “b5 e4 5c 98 5a 7e 05 ff f4 c6 a3 45 13 48 0b c6 9d e4 5d f5”. In Windows certificate store this is called “Thumbprint”, if “Thumbprint algorithm” is “sha1”.

2. **Certificate Issuer and SerialNumber**
   Options `-ci` and `-cno`
   - Certificate Issuer (e.g. “QV Schweiz CA”), in Windows certificate store this is called “Issued By”.
   - Serial number of the certificate (hexadecimal string representation, e.g. “4c 05 58 fb”). This is a unique number assigned to the certificate by its issuer. In Windows certificate store this is the field called “Serial number” in the certificate’s “Details” tab.

3. **Certificate Name and optionally Issuer**
   Options `-cn` and `-c1`
   - Common Name of the certificate (e.g. “PDF Tools AG”), in Windows certificate store this is called “Issued To”.
   - Optional: Certificate Issuer (e.g. “QV Schweiz CA”), in Windows certificate store this is called “Issued By”.

Using PKCS#11 stores with missing issuer certificates

Some PKCS#11 devices contain the signing certificate only. However, in order to embed revocation information it is important, that the issuer certificates, i.e. the whole trust chain, is available as well.

On Windows, missing issuer certificates can be loaded from the Windows certificate store. So the missing certificates can be installed as follows:

1. Get the certificates of the trust chain. You can download them from the website of your certificate provider or do the following:
   a. Sign a document and open the output in Adobe Acrobat
   b. Go to “Signature Properties” and then view the signer’s certificate
   c. Select a certificate of the trust chain
   d. Export the certificate as “Certificate File” (extension .cer)
   e. Do this for all certificates of the trust chain
2. Open the exported files by double clicking on them in the Windows Explorer
3. Click button “Install Certificate...”
4. Select “automatically select the certificate store based on the type of certificate” and finish import

**Cryptographic Suites**

**Message Digest Algorithm**

The default hash algorithm to create the message digest is **SHA-256**. Other algorithms can be chosen by setting the provider session property `MessageDigestAlgorithm`, for which supported values are:

- **SHA-1**  This algorithm is considered broken and therefore strongly discouraged by the cryptographic community.
- **SHA-256** (default)
- **SHA-384**
- **SHA-512**
- **RIPEMD-160**

**Signing Algorithm**

The signing algorithm can be configured by setting the provider session property `SigAlgo`. Supported values are:

- **RSA_RSA** (default)  This is the RSA PKCS#1v1.5 algorithm which is widely supported by cryptographic providers.
- **RSA_SSA_PSS**  This algorithm is sometimes also called RSA-PSS.

Signing will fail if the algorithm is not supported by the cryptographic hardware. The device must support either the signing algorithm `CKM_RSA_PKCS_PSS` (i.e. RSA_SSA_PSS) or `CKM_RSA_X_509` (i.e. raw RSA).

**Note:** Setting the signing algorithm only has an effect on signatures created by the cryptographic provider itself. All signed data acquired from external sources might use other signing algorithms, specifically the issuer signatures of the trust chain, the time-stamp's signature, or those used for the revocation information (CRL, OCSP). It is recommended to verify, that the algorithms of all signatures provide a similar level of security.

**5.5.2 Windows Cryptographic Provider**

This provider uses Windows infrastructure to access certificates and to supply cryptographic algorithms. Microsoft Windows offers two different APIs, the Microsoft CryptoAPI and Cryptography API Next Generation (CNG).

- **Microsoft CryptoAPI**  Provides functionality for using cryptographic algorithms and for accessing certificates stored in the Windows certificate store and other devices, such as USB tokens, with Windows integration.

  Microsoft CryptoAPI does not support some new cryptographic algorithms, such as SHA-256.

- **Cryptography API: Next Generation (CNG)**  CNG is an update to CryptoAPI. It extends the variety of available cryptographic algorithms, e.g. by the SHA-256 hashing algorithms. If possible the 3-Heights™ PDF to PDF/A Converter Service performs cryptographic calculations with CNG instead of CryptoAPI.

  CNG is available only if:

  - The operating system is at least Windows Vista or Windows Server 2008.
  - The provider of the signing certificate’s private key, e.g. the USB Token or SmartCard, supports CNG.
If CNG is not available, the CryptoAPI's cryptographic algorithms are used. In any case, CryptoAPI is used for the certificate accessing functionalities.

**Certificate Store:** It is important, that the service has access to the certificate store that contains the signing certificate. For instance, with the default configuration a service does not have access to the users' certificate store. See Store Location for more information on the topic.

**Default Message Digest Algorithm:** Since version 4.6.12.0 of the 3-Heights™ PDF to PDF/A Converter Service, the default message digest algorithm is SHA-256. As a result, signing will fail if CNG is not available (error message “Private key not available.”). To use SHA-1, the provider session property MessageDigestAlgorithm can be used. Note that the use of SHA-1 is strongly discouraged by the cryptographic community.

### Configuration

**Provider** Option `-cp`

The provider configuration string has the following syntax:

```
'[‹ProviderType›:]‹Provider›[;‹PIN›]
```

The ‹ProviderType› and ‹PIN› are optional. The corresponding drivers must be installed on Windows. If CNG is available, ‹ProviderType› and ‹Provider› are obsolete and can be omitted.

Optionally, when using an advanced certificate, the pin code (password) can be passed as an additional, semicolon separated parameter ‹PIN›. This does not work with qualified certificates, because they always require the pin code to be entered manually and every time.

If ‹Provider› is omitted, the default provider is used. The default provider is suitable for all systems where CNG is available.

**Examples:** Use the default provider with no pin.

```
Provider = ""
```

**Examples:** “123456” being the pin code.

```
Provider = ";123456"
```

```
Provider = "Microsoft Base Cryptographic Provider v1.0;123456"
```

```
Provider = "PROV_RSA_AES:Microsoft Enhanced RSA and AES Cryptographic" \
  + "Provider;123456"
```

**Certificate Store**  Option `-csn`
The value for the certificate store depends on the OS. Supported values are: “CA”, “MY” and “ROOT”. For signature creation the default store “MY” is usually the right choice.

**Store Location** Option `-csl`

Either of the following store locations
- “Local Machine”
- “Current User” (default)

Usually personal certificates are stored in the “Current User” location and company-wide certificates are stored under “Local Machine”.

In order to use a signing certificate from a user’s certificate store, the service must be logged on as the user:
1. Open the “Services” window of the Control Panel
2. Open the “Properties” dialog of the 3-Heights™ PDF to PDF/A Converter Service
3. Go to the “Log On” tab and enter the credentials of the user

Note that some cryptographic hardware (such as smart cards or USB Tokens) require an interactive environment. As a result, the private key might not be available, unless the 3-Heights™ PDF to PDF/A Converter Service is run interactively.

Certificates in the store “Local Machine” are available to all users. However, in order to sign a document, you need access to the signing certificate’s private key. The private key is protected by Windows ACLs and typically readable for Administrators only. Use the Microsoft Management Console (mmc.exe) in order to grant access to the private key for other users as follows: Add the Certificates Snap-in for the certificates on Local Machine. Right-click on the signing certificate, click on “All Tasks” and then “Manage Private Keys...” where you can set the permissions.

**Selecting a Certificate for Signing**

Within the certificate store selected by **Store Location** and **Certificate Store** the selection of the signing certificate works the same as with the PKCS#11 provider, which is described here: [Selecting a Certificate for Signing](#)

**Certificates**

In order to sign a PDF document, a valid, existing certificate name must be provided and its private key must be available.

There are various ways to create or obtain a certificate. How this is done is not described in this document. This document describes the requirements for, and how to use the certificate.

On the Windows operating system certificates can be listed by the Microsoft Management Console (MMC), which is provided by Windows. In order to see the certificates available on the system, do the following steps:
1. To launch the MMC, go to Start → Run…→ type “mmc”, or start a Command Prompt and type “mmc”.

![Run](image.png)
2. Under “File” → “Add/Remove Snap-in”
3. Choose “Certificates” and click the “Add” button
4. In the next window choose to manage certificates for “My user account”
5. Click “Finish”
6. The certificate must be listed under the root “Certificates - Current User”, for example as shown in the screenshot below:

7. Double-click the certificate to open. The certificate name corresponds to the value “Issued to”.

8. In the tab Detail of the certificate, there is a field named “Key Usage”. This field must contain the value “Digital Signature”. Additional values are optional, see also screenshot.
You must have the private key that corresponds to this certificate.

Qualified Certificates

A qualified certificate can be obtained from a certificate authority (CA). Besides the requirements listed in the previous chapter it has the additional requirement to contain the key “Authority Information Access” which contains the information about the OCSP server.
Cryptographic Suites

The message digest algorithm as well as the signing algorithm can be chosen as described for the PKCS#11 provider in Cryptographic Suites.

The `MessageDigestAlgorithm` can only be set to a value other than `SHA-1` if the private key's provider supports CNG.

The `SigAlgo` can only be set to `RSA_SSA_PSS` if the private key's provider supports CNG.

### 5.5.3 SwissSign Digital Signing Service

**Provider** Option `-cp`

The provider configuration string contains the URL to the service endpoint.

**Provider Configuration** The provider can be configured using provider session properties.

There are two types of properties:

- **“String” Properties:** String properties are set using option `-cps`.
- **“File” Properties:** File properties are set using option `-cpf`.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Identity</strong></td>
<td>String</td>
<td>required</td>
<td>The identity of your signing certificate. Example: <code>My Company:Signing Cert 1</code></td>
</tr>
<tr>
<td><strong>DSSProfile</strong></td>
<td>String</td>
<td>required</td>
<td>Must be set to <code>http://dss.swisssign.net/dss/profile/pades/1.0</code></td>
</tr>
<tr>
<td><strong>SSLClientCertificate</strong></td>
<td>File</td>
<td>required</td>
<td>SSL client certificate in PKCS#12 Format (<code>.p12</code>, <code>.pfx</code>). File must contain the certificate itself, all certificates of the trust chain and the private key.</td>
</tr>
<tr>
<td><strong>SSLClientCertificatePassword</strong></td>
<td>String</td>
<td>optional</td>
<td>Password to decrypt the private key of the SSL client certificate.</td>
</tr>
<tr>
<td><strong>SSLServerCertificate</strong></td>
<td>File</td>
<td>recommended</td>
<td>Certificate of the server or its issuer (CA) certificate (<code>.crt</code>). The certificate may be in either PEM (ASCII text) or DER (binary) form. Note: If this property is not set, the server certificate's trustworthiness cannot be determined. As a result, the connection is not guaranteed to be secure.</td>
</tr>
</tbody>
</table>
RequestID  | String  | recommended | Any string that can be used to track the request.  
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Example: An UUID like AE57F021-C0EB-4AE0-8E5E-67FB93E5BC7F</td>
</tr>
</tbody>
</table>

**Signature Configuration**  
The signature can be customized using standard options of the 3-Heights™ PDF to PDF/A Converter Service.

<table>
<thead>
<tr>
<th>Description</th>
<th>Required</th>
<th>Value</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Name</td>
<td>required</td>
<td>The name of the signer must be set.</td>
<td>Option -cn.</td>
</tr>
<tr>
<td>Time-stamp</td>
<td>optional</td>
<td>Use the value urn:ietf:rfc:3161 to embed a time-stamp.</td>
<td>Option -tsu</td>
</tr>
<tr>
<td>Signature Format</td>
<td>optional</td>
<td>To set the signature format</td>
<td>Option -st.  Must be adbe.pkcs7.detached</td>
</tr>
<tr>
<td>Revocation Info</td>
<td>recommended</td>
<td>To embed OCSP responses or CRL.</td>
<td>Option -co4</td>
</tr>
<tr>
<td>Visual Appearance</td>
<td>optional</td>
<td>See separate chapter How to Create a Visual Appearance of a Signature.</td>
<td></td>
</tr>
</tbody>
</table>

**Proxy Configuration**  
If a proxy is used for the connection to the service, see chapter How to Use a Proxy for more information.

### 5.5.4 QuoVadis sealsign

**Provider**  
Option `-cp`

The provider configuration string contains the URL to the QuoVadis sealsign service.

- Demo service:  
  https://services.sealsignportal.com/sealsign/ws/BrokerClient  
- Productive service:  
  https://qvchsvsws.quovadisglobal.com/sealsign/ws/BrokerClient

**Provider Configuration**  
The provider can be configured using provider session properties that can be set using the options `-cps` or `-cps`.

---

3 This parameter is not used for certificate selection, but for the signature appearance and signature description in the PDF only.
4 The recommendation is to not use the option `-co`.  

© PDF Tools AG – Premium PDF Technology  
3-Heights™ PDF to PDF/A Converter Service, April 21, 2020 | 47/90
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity</td>
<td>String</td>
<td>required</td>
<td>The account ID is the unique name of the account specified on the server. Example: Rigora</td>
</tr>
<tr>
<td>Profile</td>
<td>String</td>
<td>required</td>
<td>The profile identifies the signature specifications by a unique name. Example: Default</td>
</tr>
<tr>
<td>secret</td>
<td>String</td>
<td>required</td>
<td>The secret is the password which secures the access to the account. Example: Ne=EKEd33FeCk70</td>
</tr>
<tr>
<td>clientId</td>
<td>String</td>
<td>required</td>
<td>A client ID can be used to help separating access and creating better statistics. If specified in the account configuration it is necessary to provide this value. Example: 3949-4929-3179-2818</td>
</tr>
<tr>
<td>pin</td>
<td>String</td>
<td>required</td>
<td>The PIN code is required to activate the signing key. Example: 123456</td>
</tr>
</tbody>
</table>

**Signature Configuration** The signature can be customized using standard options.

<table>
<thead>
<tr>
<th>Description</th>
<th>Required</th>
<th>Value</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Name</td>
<td>required</td>
<td>The name of the signer must be set. Option -cn.</td>
<td></td>
</tr>
<tr>
<td>Time-stamp</td>
<td>-</td>
<td>Not available.</td>
<td></td>
</tr>
<tr>
<td>Revocation Info</td>
<td>recommended</td>
<td>To embed OCSP responses or CRL. Option -co6</td>
<td></td>
</tr>
<tr>
<td>Visual Appearance</td>
<td>optional</td>
<td>See separate chapter How to Create a Visual Appearance of a Signature.</td>
<td></td>
</tr>
</tbody>
</table>

**Proxy Configuration** If a proxy is used for the connection to the service, see chapter How to Use a Proxy for more information.

---

5 This parameter is not used for certificate selection, but for the signature appearance and signature description in the PDF only.

6 The recommendation is to not use the option -co.
5.5.5 Swisscom All-in Signing Service

General Properties

To use the signature service, the following general properties have to be set:

<table>
<thead>
<tr>
<th>Description</th>
<th>Required</th>
<th>Value</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Name</td>
<td>required</td>
<td>Name of the signer(^7).</td>
<td>Option  <code>-cn</code></td>
</tr>
<tr>
<td>Provider</td>
<td>required</td>
<td>The service endpoint URL of the REST service. Example: <a href="https://ais.swisscom.com/AIS-Server/rs/v1.0/sign">https://ais.swisscom.com/AIS-Server/rs/v1.0/sign</a></td>
<td>Option  <code>-cp</code></td>
</tr>
<tr>
<td>Time-stamp</td>
<td>optional</td>
<td>Use the value urn:ietf:rfc:3161 to embed a time-stamp.</td>
<td>Option  <code>-tsu</code></td>
</tr>
<tr>
<td>Signature Format</td>
<td>optional</td>
<td>To set the signature format. Supported values are adbe.pkcs7.detached, ETSI.CAdES.detached.</td>
<td>Option  <code>-st</code></td>
</tr>
<tr>
<td>Revocation Info</td>
<td>optional</td>
<td>To embed OCSP responses. Supported with adbe.pkcs7.detached only.</td>
<td>Option  <code>-co</code></td>
</tr>
</tbody>
</table>

If a proxy is used for the connection to the service, see chapter How to Use a Proxy for more information.

Provider Session Properties

In addition to the general properties, a few provider specific session properties have to be set.

There are two types of properties:

- **“String” Properties:** String properties are set using option `-cps`.
- **“File” Properties:** File properties are set using option `-cpf`.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSSProfile</td>
<td>String</td>
<td>required</td>
<td>Must be set to <a href="http://ais.swisscom.ch/1.0">http://ais.swisscom.ch/1.0</a></td>
</tr>
</tbody>
</table>

\(^7\) This parameter is not used for certificate selection, but for the signature appearance and signature description in the PDF only.
### SSLClientCertificate
- **Type**: File
- **Required**: required
- **Value**: SSL client certificate in PKCS#12 Format (.p12, .pfx).
  File must contain the certificate itself, all certificates of the trust chain and the private key.

### SSLClientCertificatePassword
- **Type**: String
- **Required**: optional
- **Value**: Password to decrypt the private key of the SSL client certificate.

### SSLServerCertificate
- **Type**: File
- **Recommended**: recommended
- **Value**: Certificate of the server or its issuer (CA) certificate (.crt). The certificate may be in either PEM (ASCII text) or DER (binary) form.
  Note: If this property is not set, the server certificate's trustworthiness cannot be determined. As a result, the connection is not guaranteed to be secure.

### Identity
- **Type**: String
- **Required**: required
- **Value**: The Claimed Identity string as provided by Swisscom:
  `<customer name>:<key identity>`

### RequestID
- **Type**: String
- **Recommended**: recommended
- **Value**: Any string that can be used to track the request.
  Example: An UUID like AE57F021-C0EB-4AE0-8E5E-67FB93E5BC7F

### On-Demand Certificates
To request an on-demand certificate, the following additional property has to be set:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SwisscomAllInOnDemandDN</td>
<td>String</td>
<td>required</td>
<td>The requested distinguished name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Example: <code>cn=Hans Muster,o=ACME,c=CH</code></td>
</tr>
</tbody>
</table>

### Step-Up Authorization using Mobile-ID
To use the step-up authorization, the following additional properties have to be set:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SwisscomAllInMSISDN</td>
<td>String</td>
<td>required</td>
<td>Mobile phone number.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Example: <code>+41798765432</code></td>
</tr>
<tr>
<td>SwisscomAllInMessage</td>
<td>String</td>
<td>required</td>
<td>The message to be displayed on the mobile phone.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Example: <code>Pipapo halolu.</code></td>
</tr>
</tbody>
</table>
Those properties have to comply with the Swisscom Mobile-ID specification.

### 5.5.6 GlobalSign Digital Signing Service

**Provider**  Option `-cp`

The provider configuration string contains the URL to the service endpoint.

`https://emea.api.dss.globalsign.com:8443/v2`

**Provider Configuration**  The provider can be configured using provider session properties.

There are two types of properties:

- **"String" Properties:**
  
  String properties are set using option `-cps`.

- **"File" Properties:**
  
  File properties are set using option `-cpf`.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>api_key</td>
<td>String</td>
<td>required</td>
<td>Your account credentials' key parameter for the login request.</td>
</tr>
<tr>
<td>api_secret</td>
<td>String</td>
<td>required</td>
<td>Your account credentials' secret parameter for the login request.</td>
</tr>
<tr>
<td>Identity</td>
<td>String</td>
<td>required</td>
<td>Parameter to create the signing certificate.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Example for an account with a static identity: {}</td>
</tr>
</tbody>
</table>
|                    |        |          | Example for an account with a dynamic identity: 
|                    |        |          | { "subject_dn": { "common_name": "John Doe" } }                     |
| SSLClientCertificate | File   | required | SSL client certificate in PKCS#12 Format (.p12, .pfx).                |
|                    |        |          | File must contain the certificate itself, all certificates of the trust chain and the private key. |
| SSLClientCertificatePassword | String | optional | Password to decrypt the private key of the SSL client certificate.   |
**SSLServerCertificate**

<table>
<thead>
<tr>
<th>Description</th>
<th>Required</th>
<th>Value</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSLServerCertificate</td>
<td>File</td>
<td>recommended Certificate of the server or its issuer (CA) certificate (.crt). The certificate may be in either PEM (ASCII text) or DER (binary) form. Note: If this property is not set, the server certificate's trustworthiness cannot be determined. As a result, the connection is not guaranteed to be secure.</td>
<td></td>
</tr>
</tbody>
</table>

**Signature Configuration**

The signature can be customized using standard options of the 3-Heights™ PDF to PDF/A Converter Service.

<table>
<thead>
<tr>
<th>Description</th>
<th>Required</th>
<th>Value</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Name</td>
<td>required</td>
<td>The name of the signer must be set.</td>
<td>Option -cn.</td>
</tr>
<tr>
<td>Time-stamp</td>
<td>recommended</td>
<td>Use the value urn:ietf:rfc:3161 to embed a time-stamp.</td>
<td>Option -tsu</td>
</tr>
<tr>
<td>Signature Format</td>
<td>optional</td>
<td>To set the signature format</td>
<td>Option -st.</td>
</tr>
<tr>
<td>Revocation Info</td>
<td>recommended</td>
<td>To embed OCSP responses or CRL.</td>
<td>Option -co²</td>
</tr>
<tr>
<td>Visual Appearance</td>
<td>optional</td>
<td>See separate chapter How to Create a Visual Appearance of a Signature.</td>
<td></td>
</tr>
</tbody>
</table>

**Proxy Configuration**

If a proxy is used for the connection to the service, see chapter How to Use a Proxy for more information.

**How to create the SSL client certificate**

When creating a new account, GlobalSign will issue an SSL client certificate clientcert.crt. The following command creates a PKCS#12 file certificate.p12 that can be used for the SSLClientCertificate:

```bash
openssl pkcs12 -export -out certificate.p12 -inkey privateKey.key -in clientcert.crt
```

**How to get the SSL server certificate**

The SSL server certificate can either be found in the technical documentation of the "Digital Signing Service" or downloaded from the server itself:

1. Get the server’s SSL certificate:

---

8 This parameter is not used for certificate selection, but for the signature appearance and signature description in the PDF only.

9 The recommendation is to not use the option --co.
openssl s_client -showcerts -connect emea.api.dss.globalsign.com:8443 ^
-certificate clientcert.crt -key privateKey.key

2. The certificate is the text starting with “-----BEGIN CERTIFICATE-----” and ending with “-----END CERTIFICATE-----”. Use the text to create a text file and save it as server.crt.

3. Use server.crt or one of its CA certificates for the SSLServerCertificate.

Advice on using the service

Note that there are rate limits for both creating new identities and for signing operations. So, if multiple documents must be signed at once, it is advisable to use the API version of the 3-Heights™ PDF to PDF/A Converter Service, which can re-use the same session (and hence its signing certificates) for signing.

Due to the short-lived nature of the signing certificates, it is important to embed revocation information immediately. For example by using the option -dss of the 3-Heights™ PDF Security Shell or not -co. Furthermore it is highly recommended to embed a time-stamp in order to prove that the signature was created during the certificate's validity period.

5.6 How to Create Digital Signatures

This chapter describes the steps that are required to create different types of digital signatures. A good introductory example can be found in the chapter How to Create Electronic Signatures.

5.6.1 How to Create a PAdES Signature

The PAdES European Norm (ETSI EN 319 142) recommends to use one of the following four baseline signature levels.

- **PAdES-B-B** A digital signature.
- **PAdES-B-T** A digital signature with a time-stamp token.
- **PAdES-B-LT** A digital signature with a time-stamp token and signature validation data. The signature is a long-term signature or “LTV enabled”.
- **PAdES-B-LTA** A digital signature with a time-stamp token and signature validation data protected by a document time-stamp.

The lifecycle of digital signatures in general and usage these signature levels in particular are described in more detail in chapter 8.11.6 “Digital signatures lifecycle” of ETSI TR 119 110.

Note: The Decision 2015/1506/EU of the eIDAS Regulation (Regulation (EU) N°910/2014) still refers to the previous legacy PAdES baseline signature standard ETSI TS 103 172. However, the signatures as created by the 3-Heights™ PDF to PDF/A Converter Service are compatible.

The Compatibility of PAdES Signature Levels shows, to which other standards the signature levels described above and as created by the 3-Heights™ PDF to PDF/A Converter Service conform.
Compatibility of PAdES Signature Levels

<table>
<thead>
<tr>
<th>ETSI EN 319 142</th>
<th>ETSI TS 102 778</th>
<th>ETSI TS 103 172</th>
<th>ISO 14533-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAdES-B-B</td>
<td>PAdES-BES (Part 3)</td>
<td>PAdES B-Level</td>
<td>-</td>
</tr>
<tr>
<td>PAdES-B-T</td>
<td>PAdES-BES (Part 3)</td>
<td>PAdES T-Level</td>
<td>PAdES-T</td>
</tr>
<tr>
<td>PAdES-B-LT</td>
<td>PAdES-BES (Part 3)</td>
<td>PAdES LT-Level</td>
<td>PAdES-A</td>
</tr>
<tr>
<td>PAdES-B-LTA</td>
<td>PAdES-LTV (Part 4)</td>
<td>PAdES LTA-Level</td>
<td>PAdES-A</td>
</tr>
</tbody>
</table>

Requirements

For general requirements and preparation steps see chapter [How to Create Electronic Signatures](#).

<table>
<thead>
<tr>
<th>Level</th>
<th>Signing Certificate</th>
<th>Time-stamp</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAdES-B-B</td>
<td>any</td>
<td>no</td>
<td>3-Heights™ PDF to PDF/A Converter Service</td>
</tr>
<tr>
<td>PAdES-B-T</td>
<td>any</td>
<td>required</td>
<td>3-Heights™ PDF to PDF/A Converter Service</td>
</tr>
<tr>
<td>PAdES-B-LT</td>
<td>advanced or qualified certificate</td>
<td>required</td>
<td>3-Heights™ PDF Security</td>
</tr>
<tr>
<td>PAdES-B-LTA</td>
<td>advanced or qualified certificate</td>
<td>required</td>
<td>3-Heights™ PDF Security</td>
</tr>
</tbody>
</table>

Make sure the trust store of your cryptographic provider contains all certificates of the trust chain, including the root certificate. Also include the trust chain of the time-stamp signature, if your TSA server does not include them in the time-stamp.

**Note on linearization:** Because signature levels PAdES-B-LT and PAdES-B-LTA must be created in a two-step process, the files cannot be linearized. When creating signature levels PAdES-B-B or PAdES-B-T that might later be augmented, linearization should not be used.

**PAdES vs. CAdES:** CAdES is an ETSI standard for the format of digital signatures. The format used in PAdES is based on CAdES, which is why the format is called `ETSI.CAdES.detached` (see `-st`). Because PAdES defines additional requirements suitable for PDF signatures, mere CAdES conformance is not sufficient.

Create a PAdES-B-B Signature

**Input Document**  Any PDF document.

**Cryptographic Provider**  A cryptographic provider that supports the creation of PAdES signatures.
**5.6.2 How to Create a Visual Appearance of a Signature**

Each signature may have a visual appearance on a page of the document. The visual appearance is optional and has no effect on the validity of the signature. Because of this and because a visual appearance may cover important content of the page, the 3-Heights™ PDF to PDF/A Converter Service creates invisible signatures by default.

In order to create a visual appearance, a non-empty signature rectangle must be set. For example, by setting the option `-ar 10 10 200 50` the following appearance is created:

![Signature Appearance Example]

Different properties of the visual appearance can be specified.

**Page and Position**  See options `-ap` and `-ar`.

**Text**  Two text fragments can be set using two different fonts, font sizes, and colors see options `-at1`, `-at2`, `-af1`, `-af2`, `-afs1`, and `-afs2`.

**Background image**  See options `-abg`.

---

**5.6.3 Miscellaneous**

**Caching of CRLs, OCSP, and Time-stamp Responses**

In order to improve the speed when mass signing, the 3-Heights™ PDF to PDF/A Converter Service provides a caching algorithm to store CRL (Certificate Revocation List), OCSP (Online Certificate Status Protocol), TSP (Time-stamp Protocol) and data from signature services. This data is usually valid over period of time that is defined by the protocol, which is normally at least 24 hours. Caching improves the speed, because there are situations when the server does not need to be contacted for every digital signature.

The following caches are stored automatically by the 3-Heights™ PDF to PDF/A Converter Service at the indicated locations within the Cache Directory:
The caches can be cleared by deleting the files. Usage of the caches can be deactivated by setting the option `-nc`.

The files are automatically updated if the current date and time exceeds the "next update" field in the OCSP or CRL response respectively or the cached data was downloaded more than 24 hours ago.

**How to Use a Proxy**

The 3-Heights™ PDF to PDF/A Converter Service can use a proxy server for all communication to remote servers, e.g. to download CRL or for communication to a signature service. The proxy server can be configured using the provider session property `Proxy`. The property's value must be a string with the following syntax:

```
http[s]://[<user>[:<password>@]<host>[:<port>]]
```

Where:
- `<user>[:<password>]` (optional): Credentials for connection to proxy (basic authorization).
- `<host>`: Hostname of proxy.
- `<port>`: Port for connection to proxy.

For SSL connections, e.g. to a signature service, the proxy must allow the HTTP CONNECT request to the signature service.

**Example:** Configuration of a proxy server that is called "myproxy" and accepts HTTP connections on port 8080.

```
-cps "Proxy" "http://myproxy:8080"
```

**Configuration of Proxy Server and Firewall**

For the application of a time-stamp or online verification of certificates, the signature software requires access to the server of the certificates' issuer (e.g. http://ocsp.quovadisglobal.com or http://platinum-qualified-g2.ocsp.swisssign.net/) via HTTP. The URL for verification is stored in the certificate; the URL for time-stamp services is provided by the issuer. In case these functions are not configured, no access is required.

In organizations where a web proxy is used, it must be ensured that the required MIME types are supported. These are:

**OCSP**
- `application/ocsp-request`
- `application/ocsp-response`

---

The sizes of the time-stamp responses are cached only. Cached Time stamp responses cannot be embedded but used for the computation of the signature length only.
**Time-stamp**

- application/timestamp-query
- application/timestamp-reply

**Signature services**

- Signature service specific MIME types.

### Setting the Signature Build Properties

In the signature build properties dictionary the name of the application that created the signature can be set using the provider session properties `Prop_Build.App.Name` and `Prop_Build.App.REx`. The default values are “3-Heights™ PDF to PDF/A Converter Service” and its version.

### 5.7 How to Validate Digital Signatures

#### 5.7.1 Validation of a Qualified Electronic Signature

There are basically three items that need to be validated:

1. Trust Chain
2. Revocation Information (optional)
3. Time-stamp (optional)

Validation can be in different ways, e.g. Adobe Acrobat, from which the screenshots below are taken.

**Trust Chain**

Before the trust chain can be validated, ensure the root certificate is trusted. There are different ways to add a certificate as trusted root certificate. The best way on Windows is this:

1. Retrieve a copy of the certificate containing a public key. This can be done be requesting it from the issuer (your CA) or by exporting it from an existing signature to a file (`CertExchange.cer`). Ensure you are not installing a malicious certificate!
2. Add the certificate to the trusted root certificates. If you have the certificate available as file, you can simply double-click it to install it.

After that you can validate the signature, e.g. by open the PDF document in Adobe Acrobat, right-click the signature and select “Validate”, then select “Properties” and select the tab “Trust”. There the certificate should be trusted to “sign documents or data”. 
Revocation Information

An OCSP response or CRL must be available. This is shown in the tab “Revocation”. The details should mention that “the certificate is considered valid”.

The presence of revocation information must be checked for the signing certificate and all certificates of its trust chain except for the root certificate.
**Time-stamp**

The signature can optionally contain a time-stamp. This is shown in the tab “Date/Time”. The certificate of the time-stamp server must also be trusted, i.e. its trust chain should be validated as described in the section Trust Chain above.
5.7.2 Validation of a PAdES LTV Signature

Verifying if a signature conforms to the PAdES LTV standard is similar to validating a Qualified Electronic Signature. The following must be checked:

1. Trust Chain
2. Revocation information
3. Time-stamp
4. LTV expiration date
5. Other PAdES Requirements

Trust Chain

Trust chain validation works the same as for validating Qualified Electronic Signatures.

Revocation Information

Revocation information (OCPS response or CRL) must be valid and embedded into the signature. In the details, verify that the revocation check was performed using data that was “was embedded in the signature or embedded in the document”. Revocation information that “was contained in the local cache” or “was requested online” is not embedded into the signature and does not meet PAdES LTV requirements. If Adobe Acrobat claims that revocation
information is contained in the local cache, even though it is embedded into the document, restart Adobe Acrobat and validate the signature again.

**Time-stamp**

A time-stamp must be embedded and validated as described for validating Qualified Electronic Signatures. If a document contains multiple time-stamps, all but the latest one must contain revocation information.

**LTV Expiration Date**

The long term validation ability expires with the expiration of the signing certificate of the latest time-stamp.

The life-time of the protection can be further extended beyond the life-of the last time-stamp applied by adding further DSS information to validate the previous last time-stamp along with a new time-stamp. This process is described in chapter [How to Create a PAdES Signature](#).

**Other PAdES Requirements**

Certain other PAdES requirements, such as requirements on the PKCS#7 CMS, cannot be validated using Adobe Acrobat. For this, use the 3-Heights™ PDF Security API for validation.
6 Interface Reference

6.1 Service Control Commands

These options are used to control the service. The create and delete functions require administrator rights. The start and stop functions require operator rights.

6.1.1 -a Pause Service

Pause Service  -a

This option pauses the service.

Pdf2PdfSvr  -a

6.1.2 -c Create Service

Create Service  -c

The 3-Heights™ PDF to PDF/A Converter Service is created using the option -c.

Pdf2PdfSvr  -c

Important: It is essential that Pdf2PdfSvr.exe is on a non-mapped drive.

6.1.3 -d Delete Service

Delete Service  -d

The 3-Heights™ PDF to PDF/A Converter Service can be deleted with the option -d. It is best used after the service has already been stopped.

Pdf2PdfSvr  -d

6.1.4 -i List the Usage

List the Usage  -i

The option -i lists the current version and date of the service along with all available settings.

Pdf2PdfSvr  -i
### 6.1.5 -o Continue Service

**Continue Service**

This option resumes the service.

Pdf2PdfSvr -o

### 6.1.6 -q Query Current Status of Service

**Query Current Status of Service**

This option returns the current status of the service.

Pdf2PdfSvr -q

The service starts automatically during system startup.

The service is stopped.


### 6.1.7 -s Start Service

**Start Service**

Once created, the 3-Heights™ PDF to PDF/A Converter Service can be started with the option -s.

Pdf2PdfSvr -s

### 6.1.8 -t Stop Service

**Stop Service**

To stop the service, use the option -t.

Pdf2PdfSvr -t

If "stop" is called while the service is "running", the current job (all pages) will be finished, after that the service is stopped.

If the service was "paused" before calling "stop", the current page will be finished processing. After that page, the job is aborted.

### 6.1.9 -x Run as Executable

**Run as Executable**
With this option, the PDF to PDF/A Converter Service runs as an executable instead of as a Windows Service. It provides the same functionality as long as the executable is "running".

```
Pdf2PdfSvr -x
```

## 6.2 Configuration Options

### 6.2.1 Configuration File Pdf2PdfSvr.ini

The Pdf2PdfSvr.ini configuration file defines the setting for the watched folders. It is read upon starting the service.

<table>
<thead>
<tr>
<th>Section</th>
<th>Required</th>
<th>Option</th>
<th>Description and Default Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Pdf2PdfSvr]</td>
<td>required</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AutoDelete=...</td>
<td>optional</td>
<td>true or false</td>
<td></td>
</tr>
<tr>
<td>AutoDeleteAll=...</td>
<td>optional</td>
<td>true or false</td>
<td></td>
</tr>
<tr>
<td>LogPath=...</td>
<td>optional</td>
<td>Either a path like C:\mypath\log or the keyword EventLog</td>
<td></td>
</tr>
<tr>
<td>PollingInterval=...</td>
<td>optional</td>
<td>Value in milliseconds, default 1000</td>
<td></td>
</tr>
<tr>
<td>JobPrefix=...</td>
<td>optional</td>
<td>true or false</td>
<td></td>
</tr>
<tr>
<td>LogLevel=...</td>
<td>optional</td>
<td>0 or 1</td>
<td></td>
</tr>
<tr>
<td>Threads=n</td>
<td>required</td>
<td>The number of threads</td>
<td></td>
</tr>
<tr>
<td>Thread1=-w ...</td>
<td>required</td>
<td>Options for the first thread</td>
<td></td>
</tr>
<tr>
<td>Thread2=-w ...</td>
<td>optional</td>
<td>Options for the second thread</td>
<td></td>
</tr>
<tr>
<td>Threadn=...</td>
<td>optional</td>
<td>There must be exactly as many threads as defined in Threads=n.</td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

```
[Pdf2PdfSvr]
AutoDelete=true
LogPath=EventLog
JobPrefix=false
Threads=2
Thread1=-w C:\Pdf2PdfSvr\PDFA -cl pdfa-1a
Thread2=-w C:\Pdf2PdfSvr\PDFA-Signed -cl pdfa-1a -cn "Peter Pan"
```

### Autodelete of Successfully Processed Files

```
[Pdf2PdfSvr]
```
AutoDelete=true

To delete failed documents as well use the following setting:

[Pdf2PdfSvr]
AutoDeleteAll=true

Job Number Prefix

Every time a document is copied from the watched folder to the Jobs sub folder, it is renamed by adding a 21 character prefix containing a time-stamp of the form Job-<8 digits>-<8 digits>_. For example

Job-01C61DD4-E72E1BCE_

The job number prefix ensures that several documents with the same name can correctly be processed. Adding the prefix can be prevented with the following line in the configuration file:

[Pdf2PdfSvr]
JobPrefix=false

Logpath

Log-messages created by the service are by default written to the sub-directory log. To alter the directory, add a line similar as shown below to the configuration file:

[Pdf2PdfSvr]
LogPath=C:\path\log

Messages created by the service can be added to the system's application event log instead of written to a log file. This is achieved by adding the following line to the configuration file:

[Pdf2PdfSvr]
LogPath=EventLog

The system's application log event will then log messages similar as shown below:

- CreateService: The operation completed successfully.
- StartService: The operation completed successfully.

Note: The messages are only fully accessible while the service is created.

Otherwise a message as shown below is displayed:

- The description for Event ID (1) in Source (Pdf2PdfSvr) cannot be found. The local computer may not have the necessary registry information or message DLL files to display messages from a remote computer. The following information is part of the event: DeleteService: The operation completed successfully.
Polling Interval

The polling interval defines the time in milliseconds that the polling-thread pauses between two polls. The time passing until the same watched folder is polled again (maximum pick-up time) is: The value of PollingInterval plus the actual time it takes to poll all watched folders. The higher the polling interval, the lower the network traffic, and the longer it takes until documents are picked up.

Suggested values for the polling intervals are 1000 to 10000 milliseconds.

```plaintext
[Pdf2PdfSvr]
PollingInterval=5000
```

6.2.2 -wfi Ignore Files with Certain Extensions

**Ignore Files with Certain Extensions**  
-wfi <exts>

By default, the service tries to process all files dropped into the drop-in folder, regardless of the extension. With this option, files with certain file extensions can be ignored.

**Example:** Ignore temporary files.

```plaintext
-wfi .temp.tmp
```

6.2.3 -wfs Process only Files with Certain Extensions

**Process only Files with Certain Extensions**  
-wfs <exts>

By default, the service tries to process all files dropped into the drop directory, regardless of the extension. With this option, the processing can be restricted to a set of known file extensions.

**Example:** Restrict the processing to PDF and FDF files.

```plaintext
-wfs .pdf.fdf
```

6.2.4 -w Set the Watched Folder

**Set the Watched Folder**  
-w

Use the option -w to define the path of the watched folder. This path should not contain mapped drives, since other users (such as LocalSystem) do not recognize them. This parameter must always be the first parameter of a thread.

```plaintext
-w C:\output\watchedfolder
```
6.3 General Settings

6.3.1 -ad  Allow Downgrade of PDF/A Conformance Level

If this option is set, automatic downgrade of the PDF/A conformance level is allowed, e.g. from PDF/A-1a to PDF/A-1b.

The level is downgraded under the following conditions:

- Downgrade to level B: If a file contains text that is not extractable (i.e. missing ToUnicode information).
  Example: Downgrade PDF/A-2u to PDF/A-2b.

- Downgrade to level U (PDF/A-2 and PDF/A-3) or B (PDF/A-1): Level A requires logical structure information (“tagging”) information, so if a file contains no such information, its level is downgraded.
  Logical structure information in a PDF defines the structure of content, such as titles, paragraphs, figures, reading order, tables or articles. Logical structure elements can be “tagged” with descriptions or alternative text.
  “Tagging” allows the contents of an image to be described to the visually impaired.
  It is not possible for the 3-Heights™ PDF to PDF/A Converter Service to add meaningful tagging information. Adding tagging information without prior knowledge about the input file’s structure and content is neither possible nor allowed by the PDF/A standard. For that reason, the conformance level is automatically downgraded to level B or U.
  Example: Downgrade PDF/A-1a to PDF/A-1b.

If this option is not set and an input file cannot be converted to the requested standard, e.g. because of missing “tagging” information, the conversion is aborted and the error code 4 is returned.

6.3.2 -au  Allow Upgrade from PDF/A-1 to PDF/A-2

If this option is set, automatic upgrade of the PDF/A version is allowed. If the target standard is PDF/A-1 and a file contains elements that cannot be converted to PDF/A-1, the target standard is upgraded to PDF/A-2. This avoids significant visual differences in the output file.

For example, the following elements may lead to an automatic upgrade:

- Transparency
- Optional Content Groups (Layers)
- Real values that exceed the implementation limit of PDF/A-1
- Embedded OpenType font files
- Predefined CMap encodings in Type0 fonts

If this option is not set, the conformance is not upgraded. Depending on the value of the conversion error mask (-cem) the conversion will fail with a conversion error (return code 5).
6.3.3 -af Add Associated File

Add a file to the document's embedded files. For PDF/A-3, the embedded file is associated with an object of the document, i.e. it is an associated file.

The file is embedded as-is. Embedding files is not allowed for PDF/A-1 and restricted to PDF/A conforming files for PDF/A-2.

**Description of the Parameter \( \langle i; r; m; d; f \rangle \)**

- **i** The object to associate the embedded file with. -1 for none, 0 for document, number greater than 0 for respective page. Default: 0 for PDF/A-3 and -1 otherwise.
- **r** The relationship of the embedded file to the object associated, PDF/A-3 only. Allowed values are Source, Data, Alternative, Supplement, and Unspecified. Default: Unspecified.
- **m** Mime-type of the embedded file. Default: application/octet-stream. Other common values are application/pdf, application/xml, or application/msword.
- **d** A description of the embedded file. This is presented to the user when viewing the list of embedded files.
- **f** The path (or URL) to the file to be embedded.

```bash
-cl pdfa-3a -af "0;Source;application/msword;The source document;input.doc"
```

6.3.4 -ef Add Embedded File

Add Embedded File -ef \( \langle \text{file} \rangle \)

-ef "\text{file}" is the same as -af ";;;\text{file}".

6.3.5 -ax Add XMP Metadata

Add XMP metadata from a file. Providing a path that does not exist or an invalid XMP file results in an error.

```bash
-ax metadata.xml
```

Note that the following metadata properties may be modified by the 3-Heights™ PDF to PDF/A Converter Service:

- pdf:Producer, xmp:ModifyDate, and xmp:MetadataDate
- Properties from the PDF/A Identification (pdfaid) schema and pdf:PDFVersion

6.3.6 -ma Analyze the Input File

Analyze the Input File -ma
Analyze the input file and verify if it meets a certain conformance level. In order to get a report either the option `-rs` or `-rd` can be used in combination, otherwise only the return code is set.

An output file name must be provided, since the output name also specifies the name of the log file which is generated. However, no output PDF document is created. The analysis is similar to the analysis using the 3-Heights™ PDF Validator. However, the 3-Heights™ PDF to PDF/A Converter Service is more strict in certain issues, especially concerning those corner cases of the PDF/A ISO Standard in which a conversion is strongly advised.

### 6.3.7 -cff Embed Type 1 Fonts as CFF

<table>
<thead>
<tr>
<th>Embed Type 1 Fonts as CFF</th>
<th>-cff</th>
</tr>
</thead>
</table>

Convert Type1 (PostScript) fonts to Compact Font Format before embedding. This reduces the file size. This affects the embedding of fonts only, existing Type1 fonts of the input document will not be converted.

### 6.3.8 -mc Force Conversion even if there are Analysis Errors

<table>
<thead>
<tr>
<th>Force Conversion even if there are Analysis Errors</th>
<th>-mc</th>
</tr>
</thead>
</table>

Setting this option forces the conversion even if the input file already conforms to the requested standard.

### 6.3.9 -q Image Quality

<table>
<thead>
<tr>
<th>Image Quality</th>
<th>-q &lt;n&gt;</th>
</tr>
</thead>
</table>

Set or get the image quality index for images that use a prohibited lossy compression type and must be re-compressed.

Supported values are 1 to 100. A higher value means better visual quality at the cost of a larger file size. Recommended values range from 70 to 90. The default value is 80.

**Example:**

JPX (JPEG2000) is not allowed in PDF/A-1. If a PDF contains a JPX compressed image, its compression type must be altered. Thus the 3-Heights™ PDF to PDF/A Converter Service converts it to an image with regular JPEG compression and the image quality as defined by this switch.

### 6.3.10 -cem Mask Conversion Errors

<table>
<thead>
<tr>
<th>Mask Conversion Errors</th>
<th>-cem &lt;mask&gt;</th>
</tr>
</thead>
</table>

The conversion error mask defines which operations and conditions are not allowed and consequently cause a conversion error (return value 5).
### Values for `<mask>`

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>None (never return a conversion error)</td>
</tr>
<tr>
<td>8</td>
<td>Resolve name collisions of colorants (PDF/A-2 and PDF/A-3 only).</td>
</tr>
<tr>
<td>16</td>
<td>(default) Remove optional content groups (layers) (PDF/A-1 only).</td>
</tr>
<tr>
<td>32</td>
<td>(default) Remove transparency (PDF/A-1 only).</td>
</tr>
<tr>
<td>64</td>
<td>(default) Remove embedded files.</td>
</tr>
<tr>
<td>128</td>
<td>(default) Remove non convertible XMP metadata.</td>
</tr>
<tr>
<td>512</td>
<td>Conversion of signed document forced removal of signatures.</td>
</tr>
<tr>
<td>4096</td>
<td>(default) The input document is corrupt and should be repaired. The errors encountered are printed to the log file. Some errors can be repaired, but it is crucial to review the output file and perform the post analysis.</td>
</tr>
<tr>
<td>16384</td>
<td>Font substituted.</td>
</tr>
<tr>
<td>32768</td>
<td>Remove interactive elements such as actions or annotations.</td>
</tr>
<tr>
<td>65536</td>
<td>Remove logical structure information.</td>
</tr>
</tbody>
</table>

Add up all operations are conditions to define the conversion mask. The default is 4340. In order to accept the removal of XMP metadata, set the mask to 4212.

```
-cem 4212
```

See chapter [Conversion Errors](#) for more information on conversion errors and how they can be handled.

### 6.3.11 -cef Try to convert embedded PDF documents (PDF/A-3 only)

```
Try to convert embedded PDF documents (PDF/A-3 only)  -cef
```

By default, embedded files are copied as-is during conversion to PDF/A-3. If this option is used, the 3-Heights™ PDF to PDF/A Converter Service tries to convert embedded PDF documents to PDF/A. The converted document is embedded only, if the conversion was successful.

Note that this option is relevant for PDF/A-3 only. During conversion to PDF/A-1, all embedded files are removed, whereas the conversion to PDF/A-2 converts all embedded files to PDF/A.

### 6.3.12 -ow Optimize for the Web

```
Optimize for the Web  -ow
```

Linearize the PDF output file, i.e. optimize file for fast web access.
A linearized document has a slightly larger file size than a non-linearized file and provides the following main features:

- When a document is opened in a PDF viewer of a web browser, the first page can be viewed without downloading the entire PDF file. In contrast, a non-linearized PDF file must be downloaded completely before the first page can be displayed.
- When another page is requested by the user, that page is displayed as quickly as possible and incrementally as data arrives, without downloading the entire PDF file.

The above applies only if the PDF viewer supports fast viewing of linearized PDFs.

Note: In order to make use of a linearized PDF file, the PDF must reside as a "file" on the web-server. It must not be streamed.

The 3-Heights™ PDF to PDF/A Converter Service cannot linearize signed files. So this option should not be used if a digital signature is applied. Linearizing signed files is possible with the 3-Heights™ PDF Security product.

When enabling this option, then no PDF objects will be stored in object streams in the output PDF. For certain input documents this can lead to a significant increase of file size.

### 6.3.13 -p Read an Encrypted PDF File

**Read an Encrypted PDF File**  
\[-p \langle \text{password} \rangle\]

When the input PDF file is encrypted and has a user password set, (the password to open the PDF) the password can be provided with the option -p. If for example the user password were "userpwd", then the command would look like this:

```
-p userpwd
```

When a PDF is encrypted and the user password is not provided or is incorrect, pdf2pdf cannot decrypt and read the file. Instead it will generate the following error message:

```
Cannot open file input.pdf.
```

### 6.3.14 -rd Report Conformance Violations in Detail

**Report Conformance Violations in Detail**  
\[-rd\]

This option lists all conformance violations per page. Each violation is listed with a page number (page 0 = document level), error number, a description and a counter of how many times the error occurs. The option provides more detailed information than the summary (-rs) for both the pre and the post analysis.

**Example:**

```
-rd
- Opening file input.pdf.
- Analyzing input.pdf.
  "input.pdf", 0, 53, 0x0A09C882, "The property 'xmp:Format' is not defined in schema 'XMP Basic Schema'.", 1
```
6.3.15  \texttt{-rs} Report Conformance Violations Summary

This option gives a summary of all conformance violations. If any of the following violations is detected at least once, it is reported (once). This option provides less detailed information than the detailed list per page (\texttt{-rd}) for both the pre and the post analysis.

- The file format (header, trailer, objects, xref, streams) is corrupted.
- The document doesn't conform to the PDF reference (missing required entries, wrong value types, etc.).
- The file is encrypted and the password was not provided.
- The document contains device-specific color spaces.
- The document contains illegal rendering hints (unknown intents, interpolation, transfer and halftone functions).
- The document contains alternate information (images).
- The document contains embedded PostScript code.
- The document contains references to external content (reference XObjects, file attachments, OPI).
- The document contains fonts without embedded font programs or encoding information (CMAPs).
- The document contains fonts without appropriate character to Unicode mapping information (ToUnicode maps).
- The document contains transparency.
- The document contains unknown annotation types.
- The document contains multimedia annotations (sound, movies).
- The document contains hidden, invisible, non-viewable or non-printable annotations.
- The document contains annotations or form fields with ambiguous or without appropriate appearances.
- The document contains actions types other than for navigation (launch, JavaScript, ResetForm, etc.).
- The document's meta data is either missing or inconsistent or corrupt.
- The document doesn't provide appropriate logical structure information.
- The document contains optional content (layers).

Example:

\begin{verbatim}
-rs
- Opening file input.pdf.
- Analyzing input.pdf.
  The document contains device-specific color spaces.
  The document contains fonts without embedded font programs or encoding information (CMAPs).
  The document's meta data is either missing or inconsistent or corrupt.
\end{verbatim}

6.3.16  \texttt{-cl} Set Conformance

Set the PDF/A conformance level.
Supported values for the parameters \( <\text{conf}> \) are: pdfa-1b, pdfa-1a, pdfa-2b, pdfa-2u, pdfa-2a, pdfa-3b, pdfa-3u, pdfa-3a. The default value is pdfa-2b.

Some files cannot be converted to the conformance requested. The 3-Heights™ PDF to PDF/A Converter Service can detect this and up- or downgrade the conformance automatically. Use the options \(-au\) and \(-ad\) to allow automatic up- or downgrades.

The following example will set the conformance level to PDF/A-2u:

```
-c1 pdfa-2u
```

6.3.17 \(-fd\) Add font directory

Add font directory \(-fd\) \(<\text{dir}>\)

All fonts must be embedded in order to create a valid PDF/A. If the input file contains a font which is not embedded, the font folder is searched for a font with the same name. If such a font is found, the font is embedded.

This option can be used to add (multiple) font directories to the search path for fonts.

In addition to directories added with this option, the default font directories are always considered.

```
-fd C:\MyFonts
```

See chapter Fonts for more information on the font directories and font handling of the 3-Heights™ PDF to PDF/A Converter Service in general.

6.3.18 \(-uf\) Update the Fonts’ Unicodes

Update the Fonts’ Unicodes \(-uf\) \(<\text{file}>\)

Update the fonts’ Unicodes as specified by parameter \(<\text{file}>\). The file must contain the mapping of character codes to Unicodes for specific fonts.

6.4 Color Profiles

See the dedicated section Color Spaces for more information on the topic.

6.4.1 \(-cs\) ICC Profile for Device-Specific Color Spaces

ICC Profile for Device-Specific Color Spaces \(-cs\) \(<\text{profile}>\)

This ICC profile represents the color profile of the scanner. It is required if a color space is used that is different from color ICC profile of the output intent. Initially there is a default color profile for RGB (sRGB) and CMYK (USWebCoat-edSWOP.icc) defined in the 3-Heights™ PDF to PDF/A Converter. This switch can be used to set both, the RGB and the CMYK color profile. If an RGB color profile is passed as argument, it is set as new RGB color space. If a CMYK color is provided, it is set as new CMYK color space. If an invalid file is provided, it results in error code 3. To set the color profile for both color spaces, use the switch \(-cs\) twice.

The following command sets the standard sRGB color profile as color space:
If a required color space profile is not available, a device independent color space is created automatically.

### 6.4.2 -oi ICC Profile for Output Intent

<table>
<thead>
<tr>
<th>ICC Profile for Output Intent</th>
<th>-oi &lt;profile&gt;</th>
</tr>
</thead>
</table>

The ICC profile for the output intent describes the color profile of the device (monitor or display). An output intent is required for PDF/A compatibility when converting images that do not have an embedded color profile. If no output intent is specified, a default color profile is embedded. The default color profile is the sRGB Color Space Profile.icm. If the input document already contains an output intent, the existing output intent is kept. Providing a path that does not exist or an invalid ICC color profile file results in return code 3.

-oi "C:\Windows\system32\spool\drivers\color\sRGB Color Space Profile.icm"

### 6.5 Digital Signatures

For more information on digital signatures in general, see section Digital Signatures. For more information on how to create digital signatures, see section How to Create Digital Signatures.

#### 6.5.1 -abg Signature Background Image

<table>
<thead>
<tr>
<th>Signature Background Image</th>
<th>-abg &lt;image&gt;</th>
</tr>
</thead>
</table>

This is the background image that is added to the signature. The image is centered and scaled down proportionally to fit into the given rectangle. If the path is Nothing, or the image does not exist, the appearance's background is a filled rectangle using the colors fill color and stroke color.

In order to create a signature with the image only, set the signature texts 1 and 2 to a space " ".

#### 6.5.2 -af1 Signature Font Name 1

<table>
<thead>
<tr>
<th>Signature Font Name 1</th>
<th>-af1 &lt;font name&gt;</th>
</tr>
</thead>
</table>

This defines the font used in upper text, i.e. the text that is set by the property -at1. The font can either be specified as a path to the font file, e.g. "C:\Windows\Fonts\arial.ttf", or as a font name, such as "Times New Roman, Bold". When using a font name, the corresponding font must be present in one of the font directories described in chapter Fonts.

#### 6.5.3 -af2 Signature Font Name 2

<table>
<thead>
<tr>
<th>Signature Font Name 2</th>
<th>-af2 &lt;font name&gt;</th>
</tr>
</thead>
</table>

This is the font used in lower text, i.e. the text that is set by -at2. The option works analogously to -af1.
6.5.4 -afs1 Signature Font Size 1

This defines the font size in points used in upper text, i.e. the text that is set by the property -at1. If the font size is not specified, a default value of 16pt is used.

6.5.5 -afs2 Signature Font Size 2

This is the font size in points used in lower text, i.e. the text that is set by -at2. The option works analogously to -afs1. If the font size is not specified, a default value of 8pt is used.

6.5.6 -ap Signature Page Number

Set the page number of where the visual appearance of the digital signature should be placed. The numbers are counted starting from 1 for the first page. The default is the last page. The last page can also be set using -1 as argument.

6.5.7 -ar Signature Annotation Rectangle

Set the position and size of the digital signature annotation. The default is an invisible signature (-ar 0 0 0 0). The position is defined by the four values for the lower-left corner (x, y) and dimensions (w, h) of the rectangle. The units are PDF points (1 point = 1/72 inch, A4 = 595 x 842 points, Letter = 612 x 792 points) measured from the lower left corner of the page. If either the width or height is zero or negative, an invisible signature is created, i.e. no visible appearance is created for the signature.

Example: Create a 200 by 60 points rectangle in the upper left corner of an A4 page.

-afn "..." -ar 10 770 200 60

6.5.8 -at1 Signature Text 1

This is the upper text that is added to the signature.

If this property is not set, the signing certificate's name set with -cn is added to the upper text line of the visual signature.

See section How to Create a Visual Appearance of a Signature for more information on customizing the appearance of digital signatures.
6.5.9  -at2  Signature Text 2

| Signature Text 2  | -at2  | text |

This is the lower text that is added to the signature. The text can be multi-lined by using carriage returns. If this property is not set, a three-line text is constructed that consists of:

- A statement who applied to signature
- The reason of the signature. This can be set using -cr.
- The date

See section How to Create a Visual Appearance of a Signature for more information on customizing the appearance of digital signatures.

6.5.10 -cci  Signer contact info

| Signer contact info  | -cci  | info |

Add a descriptive text as signer contact info, e.g. a phone number. This enables a recipient to contact the signer to verify the signature. This is not required in order to create a valid signature.

6.5.11 -cfp  Certificate Fingerprint

| Certificate Fingerprint  | -cfp  | fp |

License feature: Signature

Set the hex string representation of the signer certificate's sha1 fingerprint. All characters outside the ranges 0-9, a-f and A-F are ignored. In the Microsoft Management Console, the "Thumbprint" value can be used without conversion, if the "Thumbprint algorithm" is "sha1". E.g. "b5 e4 5c 98 5a 7e 05 ff f4 c6 a3 45 13 48 0b c6 9d e4 5d f5". This property can be used to select the signer certificate for signing (see description of Cryptographic Provider).

6.5.12 -ci  Certificate Issuer

| Certificate Issuer  | -ci  | issuer |

License feature: Signature

The issuer of the certificate. The "Certificate Issuer" corresponds to the common name (CN) of the issuer. In the Windows' certificate store this corresponds to "Issued by". This property can be used to select the signer certificate for signing (see description of Cryptographic Provider).

6.5.13 -cn  Certificate Name (Subject)

| Certificate Name (Subject)  | -cn  | name |

License feature: Signature

Set the name of the certificate used to sign the document (see description of Cryptographic Provider). The name corresponds to the common name (CN) of the subject. In the Windows' certificate store this corresponds to "Issued to".
Consult chapter Digital Signatures to learn more about digital signatures in general and how to sign documents with the 3-Heights™ PDF to PDF/A Converter Service.

**Example:** Sign the document

```
-cn "Philip Renggli"
```

The signature is added on the last page of the signed document.

### 6.5.14 -cno Certificate Serial Number

<table>
<thead>
<tr>
<th>Certificate Serial Number</th>
<th>-cno &lt;serialno&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>License feature:</td>
<td>Signature</td>
</tr>
</tbody>
</table>

Set the serial number of the certificate. Specify a hex string as displayed by the “Serial number” field in the Microsoft Management Console (MMC), e.g. "49 cf 7d d1 6c a9". This property can be used to select the signer certificate for signing (see description of Cryptographic Provider).

### 6.5.15 -co Do not Embed Revocation Information

<table>
<thead>
<tr>
<th>Do not Embed Revocation Information</th>
<th>-co</th>
</tr>
</thead>
</table>

This switch inhibits the embedding of revocation information such as online certificate status response (OCSP - RFC 2560) and certificate revocation lists (CRL - RFC 3280). Revocation information is either an OCSP response or a CRL, which is provided by a validation service at the time of signing and acts as proof that at the time of signing the certificate is valid. This is useful because even when the certificates expires or is revoked at a later time, the signature in the signed document remains valid.

Embedding revocation information is optional but suggested when applying advanced or qualified electronic signatures.

This option is not supported by all cryptographic providers and never for document time-stamp signatures.

Revocation information is embedded for the signing certificate and all certificates of its trust chain. This implies that both OCSP responses and CRLs can be present in the same message.

The downsides of embedding revocation information are the increase of the file size (normally by around 20 KB) and that it requires a connection to a validation service, which delays the process of signing. For mass signing it is suggested to use the caching mechanism, see section Caching of CRLs, OCSP, and Time-stamp Responses.

Embedding revocation information requires an online connection to the CA that issues them. The firewall must be configured accordingly. In case a web proxy is used, it must be ensured the following MIME types are supported when using OCSP (not required for CRL):

```
application/ocsp-request
application/ocsp-request
```

### 6.5.16 -cp Cryptographic Provider

<table>
<thead>
<tr>
<th>Cryptographic Provider</th>
<th>-cp &lt;prov&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>License feature:</td>
<td>Signature</td>
</tr>
</tbody>
</table>

© PDF Tools AG – Premium PDF Technology
This property specifies the cryptographic provider used to create and verify signatures.
For more information on the different providers available, see the description in the respective subsection of the section **Cryptographic Provider**.

- When using the **Windows Cryptographic Provider**, the value of this property is to be set to a string with the following syntax:
  
  "[ProviderType:]Provider[;PIN]"
  
  If the name of the provider is omitted, the default provider is used.

**Examples:**  "123456" being the pin code

<table>
<thead>
<tr>
<th>Provider = &quot;Microsoft Base Cryptographic Provider v1.0;123456&quot;</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Provider = &quot;;123456&quot;</th>
</tr>
</thead>
</table>

- When using the **PKCS#11 Provider**, the value of this property is to be set to a string with the following syntax:
  
  "PathToDll;SlotId;Pin"
  
**Example:**

<table>
<thead>
<tr>
<th>Provider = &quot;\WINDOWS\system32\siecap11.dll;4;123456&quot;</th>
</tr>
</thead>
</table>

- When using any of the service providers, such as the “Swisscom All-in signing service”, the value of this property is essentially the url of the service endpoint:
  
  "http[s]://server.servicedomain.com:8080/url"

### 6.5.17 -cpf Cryptographic session property (file)

**Cryptographic session property (file) -cpf <name> <file>**

File data property for configuring cryptographic session. The supported names and values are specific to the cryptographic provider.

### 6.5.18 -cps Cryptographic session property (string)

**Cryptographic session property (string) -cps <name> <str>**

String property for configuring cryptographic session. The supported names and values are specific to the cryptographic provider.

### 6.5.19 -cr Signature Reason

**Signature Reason -cr <reason>**
Add a descriptive text about the reason why the document was signed.

**Example:** Sign the document an add a reason text.

-cn "Philip Renggli" -cr "Review and approval" -ar 10 10 200 50

The signature of the resulting output looks as shown below:

![Signature example](image)

### 6.5.20 -cs1 Certificate Store Location

**Certificate Store Location** -cs1 \{location\}

For the [Windows Cryptographic Provider](https://docs.microsoft.com/en-us/windows/security/threat-protection/certificate-services/certificate-services) this defines the location of the certificate store from where the signing certificate should be taken. Supported are:

- Local Machine.
- Current User (default).

For more information, see the detailed description of the [Windows Cryptographic Provider](https://docs.microsoft.com/en-us/windows/security/threat-protection/certificate-services/certificate-services).

### 6.5.21 -csn Certificate Store Name

**Certificate Store Name** -csn \{store\}

For the [Windows Cryptographic Provider](https://docs.microsoft.com/en-us/windows/security/threat-protection/certificate-services/certificate-services) this defines the certificate store from where the signing certificate should be taken. This depends on the operating system. The default is "MY". Other supported values are: "CA" or "ROOT".

**Example:** Use the certificate store ROOT from the Local Machine account.

- cn "..." -csn ROOT -cs1 0

### 6.5.22 -nc Disable cache for CRL and OCSP

**Disable cache for CRL and OCSP** -nc

Get or set whether to disable the cache for CRL and OCSP responses.

Using the cache is safe, since the responses are cached as long as they are valid only. The option affects both signature creation and validation.

See section on [Caching of CRLs, OCSP, and Time-stamp Reponses](https://docs.microsoft.com/en-us/windows/security/threat-protection/certificate-services/certificate-services) for more information on the caches.
6.5.23  **-st  Set Signature SubFilter**

<table>
<thead>
<tr>
<th>Set Signature SubFilter</th>
<th>-st &lt;subfilter&gt;</th>
</tr>
</thead>
</table>

The `<subfilter>` indicates the encoding of the signature. The following are common values for `<subfilter>`:

- **adbe.pkcs7.detached** (PDF 1.6) Legacy PAdES Basic (ETSI TS 102 777, Part 2) signature used for document signatures.
- **ETSI.CAdES.detached** (PDF 2.0) PAdES signature as specified by European Norm ETSI EN 319 142. This type is used for document signatures. See chapter [How to Create a PAdES Signature](#) for more information.

6.5.24  **-tsc  Time-stamp Credentials**

<table>
<thead>
<tr>
<th>Time-stamp Credentials</th>
<th>-tsc &lt;cred&gt;</th>
</tr>
</thead>
</table>

If a time-stamp server requires authentication, use this switch to provide the credentials.

**Example:** Credentials commonly have the syntax `username:password`.

```shell
-cn "..." -tsu http://mytimestamp.com -tsc username:password
```

6.5.25  **-tsu  Time-stamp URL**

<table>
<thead>
<tr>
<th>Time-stamp URL</th>
<th>-tsu &lt;url&gt;</th>
</tr>
</thead>
</table>

The URL of the trusted time-stamp server (TSA) from which a time-stamp shall be acquired. This setting is only required when applying a Qualified Electronic Signature. Applying a time-stamp requires an online connection to a time server; the firewall must be configured accordingly. In case a web proxy is used, it must be ensured the following MIME types are supported:

- `application/timestamp-query`
- `application/timestamp-reply`

6.5.26  **-wpc  Web Proxy Server Credentials**

<table>
<thead>
<tr>
<th>Web Proxy Server Credentials</th>
<th>-wpc &lt;cred&gt;</th>
</tr>
</thead>
</table>

If a web proxy server is used, and it requires authentication, use this switch and the syntax `user:password`.

**Example:** Set a web proxy server URL and use authentication.

```shell
-wpu "http://proxy.example.org" -wpc user:password
```

6.5.27  **-wpu  Web Proxy Server URL**

<table>
<thead>
<tr>
<th>Web Proxy Server URL</th>
<th>-wpu &lt;url&gt;</th>
</tr>
</thead>
</table>

© PDF Tools AG – Premium PDF Technology

---

3-Heights™ PDF to PDF/A Converter Service, April 21, 2020 | 80/90
In an organization where a web proxy server is in use, it must be ensured this web proxy server is specified. The URL is something like "http://proxy.example.org" or an IP address. For more information, see the chapter How to Use a Proxy.

6.6 OCR

In order to make use of OCR, an OCR engine must be installed. The OCR engine is provided as part of a separate product: The 3-Heights™ OCR Enterprise Add-On.

The recommended options (besides -ocr, -ocl and -ocp) are:

- For scanned documents: -oca -ocri -occs
- For born-digital documents: none

6.6.1 -oca  Rotate the image to the detected angle

Rotate the image to the detected angle  -oca

The OCR engine may detect that an image needs to be rotated in order to have the text in an up-right position. If this is the case and this switch is used then the original image is replaced by the rotated image.

6.6.2 -ocb  Convert Images to Bitonal before OCR Recognition

Convert Images to Bitonal before OCR Recognition  -ocb

Specify whether the images should be converted to bi-tonal (black and white) before OCR recognition. Enabling this feature can improve the memory consumption of the OCR process.

6.6.3 -ocbc  Embed barcodes

Embed barcodes  -ocbc

Embed the recognized barcodes in the XMP metadata.

6.6.4 -occs  Correct skew angle

Correct skew angle  -occs

Correct the skew angle of images.

This option has only an effect, if the required information is provided by the OCR engine, which depends on the type and settings of the engine.

This option might change the appearance of the page and is only recommended for simple scanned documents that consist of a single image.

Using the option for digital-born documents may destroy the page layout.
6.6.5 -ocd Resolution for OCR Recognition

Resolution for OCR Recognition  -ocd  \langle n \rangle

Resample images to target resolution before they are sent to the OCR engine. If no value is set, images are re-sampled to 300 DPI for OCR, which is the preferred resolution for most OCR engines.

6.6.6 -ocl Set OCR Language

Set OCR Language  -ocl  \langle languages \rangle

In order to optimize the performance of the OCR engine, it can be given hints what languages are used. The default language of the Abbyy FineReader 11 OCR Engine is English. This switch can only be used if the switch -ocr is set. This setting depends on the OCR engine.

The following switch set the languages to English and German:

-ocr service -ocl "English, German"

See also documentation for the 3-Heights™ OCR Add-On.

6.6.7 -ocm OCR mode

OCR mode  -ocm  \langle n \rangle

Specify behavior of the 3-Heights™ PDF to PDF/A Converter Service for files with existing OCR text. Available OCR modes are the following:

1. Only perform OCR for images without existing OCR text (default).
2. If OCR engine is active, remove old OCR text and perform OCR for all images. Hence, existing OCR text is not removed if OCR engine is not active.
3. Always remove old OCR text and, if OCR engine is active, perform OCR for all images. This can be used to strip existing, without adding new OCR text.
4. Only perform OCR if input file contains no text.

Example: Set OCR mode 2

-ocr abbyy11 -ocm 2

6.6.8 -ocp Set OCR Parameters

Set OCR Parameters  -ocp  \langle params \rangle

Using this switch, OCR engine specific parameters (key/value pairs) can be set to optimize the performance.

The following switch sets a predefined profile (i.e. a configuration setting) which is optimized for creating electronic archives with high accuracy:
6.6.9 -ocr  Load OCR Engine

If a PDF document has to be made fully text searchable even if the text is part of a raster image then the images which are contained in the PDF document must be run through an OCR engine. With this switch the user can select an OCR engine, e.g. Abbyy11, and instruct the tool to embed the recognized text as a hidden layer on top of the image. If the add-in is not found or the engine cannot be initialized (because it is not installed or the license key is not valid) then an error message is issued.

If the switch -ocr is not used, no OCR is applied.

Example: The following switch sets the OCR engine to the OCR Service

-ocr service

See also documentation for the 3-Heights™ OCR Add-On.

6.6.10 -ocri  Reembed preprocessed image

This option currently requires the -occs to be set.

The OCR engine de-skews and de-noises the input image before recognizing the characters. This option controls whether the 3-Heights™ PDF to PDF/A Converter Service should use the preprocessed image or keep the original image.

This option has only an effect, if the preprocessed image is provided by the OCR engine, which depends on the type and settings of the engine.

If this option is set, the resulting image may have a different color space, compression and size.

Since this option currently requires -occs, it is recommended only for simple scanned documents.

6.6.11 -oct  Threshold Resolution for OCR

Only images with a higher resolution than the threshold are re-sampled before OCR. The default is 400 DPI. If set to -1, no re-sampling is applied.

Example: Resample all images with a resolution of more than 300 DPI to 300 DPI:

-ocd 300 -oct 1

PredefinedProfile = DocumentArchiving_Accuracy

See also documentation for the 3-Heights™ OCR Add-On.
**Example:** Resample all images with a resolution of 400 DPI or more to 300 DPI (default):

-ocd 300 -oct 400

**Example:** Do not resample:

-oct -1

**Compatibility Note:** Initially this switch was called -ocD and then renamed to -oct to avoid confusions with the switch -ocd.
7 Log File

All steps in the diagram from chapter Process Description can write to the log file. There are three types of messages in the log file: Warnings/Information, Errors and Reports.

7.1 Warnings and Information

Describe the current process step. They do not inhibit the conversion. Prefix: -

Example:

- Opening file input.pdf
- Analyzing input.pdf
- Conformance level has been downgraded to level U.
- Performing post analysis for output.pdf.
- Post analysis for output.pdf has been successful.
- File input.pdf converted successfully.

7.2 Errors

Inhibit a successful conversion. Prefix: *

Example: The input file cannot be opened

* Cannot open file input.jpg.

Example: Distinguish critical from non-critical conversion events

Critical conversion events use the prefix * and non-critical - (see chapter Conversion Errors).

<table>
<thead>
<tr>
<th>Example</th>
<th>Distinguish critical from non-critical conversion events</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Opening file input.pdf.</td>
<td></td>
</tr>
<tr>
<td>- Analyzing input.pdf.</td>
<td></td>
</tr>
<tr>
<td>- FontSubst: Substitute font 'Arial-BlackItalic' with multiple master font.</td>
<td></td>
</tr>
<tr>
<td>- Embed font 'Verdana'.</td>
<td></td>
</tr>
<tr>
<td>- Embed font 'Verdana-Bold'.</td>
<td></td>
</tr>
<tr>
<td>* Metadata: xmp:Format :: Value removed because it is not defined in the schema description. (document metadata)</td>
<td></td>
</tr>
<tr>
<td>- Create calibrated color space substitute for DeviceGray. (content of page 1)</td>
<td></td>
</tr>
<tr>
<td>- Conversion events:</td>
<td></td>
</tr>
<tr>
<td>* Parts of the XMP metadata could not be repaired and had to be removed.</td>
<td></td>
</tr>
<tr>
<td>- Font substituted.</td>
<td></td>
</tr>
<tr>
<td>- Performing post analysis for output.pdf.</td>
<td></td>
</tr>
<tr>
<td>- Post analysis for output.pdf has been successful.</td>
<td></td>
</tr>
<tr>
<td>* Conversion errors in output.pdf.</td>
<td></td>
</tr>
</tbody>
</table>

7.3 Reports

Reports are only created if the corresponding option (Details or Summary) is selected. In detailed reports, each violation is listed with a page number (page 0 = document level), pdf object number, error code, a description, and
a counter of how many times the error occurs. In a summary report, violations that are detected at least once are reported once. Prefix: none.

**Example**: Details

```
"input.pdf", 0, 53, 0x80410604, "The key Metadata is required but missing.", 1
"input.pdf", 2, 14, 0x00418704, "The font Verdana must be embedded.", 1
"input.pdf", 1, 4, 0x03418614, "A device-specific color space (DeviceGray) without an appropriate output intent is used.", 1
```

**Example**: Summary

```
The document contains fonts without embedded font programs or encoding information (CMAPs).
The document's metadata is either missing or inconsistent or corrupt.
The document doesn't provide appropriate logical structure information.
```
# 8 Version History

## 8.1 Changes in Version 6

- **Digital Signatures**
  - **New** support to create PAdES signatures (format ETSI.CAdES.detached) with the Swisscom All-in Signing Service.
  - **Improved** embedding of revocation information (OCSP, CRL, and trust chain) to always use the document security store (DSS).
  - **Changed** the creation of signatures of format ETSI.CAdES.detached to include revocation information if `-cc` is not used and if supported by the cryptographic provider.
  - **Improved** support for new version of the GlobalSign Digital Signing Service. The service endpoint should be updated to `https://emea.api.dss.globalsign.com:8443/v2`.
  - **Improved** repair of corrupt DCT stream data.
  - **Improved** search algorithm for installed fonts: User fonts under Windows are now also taken into account.

## 8.2 Changes in Version 5

- **Digital Signatures**
  - **New** support to get CRLs using HTTPS and via HTTP redirection.
  - **New** support for ZUGFeRD 2.0 hybrid electronic invoices.
  - **Improved** conversion of transparent objects to PDF/A-1.
  - **Improved** log output for conversion events. Now all conversion events are written to the log file.
  - **New** additional supported operating system: Windows Server 2019.

## 8.3 Changes in Version 4.12

- **Introduced** license feature Signature.
- **Digital Signatures**
  - **New** support to sign OCSP requests, if required by the OCSP service.
  - **New** support for OCSP requests over HTTPS.
  - **Changed** acceptance criteria for OCSP responses that specify no validity interval (missing nextUpdate field, which is uncommon). Previously a validity interval of 24 hours has been used, now 5 minutes due to Adobe® Acrobat® compatibility.
  - **New** support for Factur-X hybrid electronic invoices.
  - **New** OCR plugin “abbyy12” for the ABBYY FineReader 12 engine.
  - **Changed** behavior when adding additional font directories: The default directories are now always considered.
  - **Improved** detection of corrupt embedded fonts, DCT streams and CMap encodings.
  - **New** HTTP proxy setting in the GUI license manager.

## 8.4 Changes in Version 4.11

- **New** font substitution for CJK (Chinese, Japanese, Korean) fonts if an exact match is missing.
- **Improved** conversion of files with optional content (layers) to minimize visual differences while also preserving all content.
- **New** support for the creation of appearance streams for free text annotations that contain rich text content.
- Digital Signatures
  - **New** ability to sign documents that are larger than 2GB (64-bit version only).
  - **New** support for reading PDF 2.0 documents.
  - **New** support for the creation of output files larger than 10GB (not PDF/A-1).
  - **Improved** search in installed font collection to also find fonts by other names than TrueType or PostScript names.
  - **Improved** font subsetting of CFF and OpenType fonts.
  - **Improved** repair of corrupt image streams.
  - **New** treatment of the DocumentID. In contrast to the InstanceID the DocumentID of the output document is inherited from the input document.
  - **New** option `-cef` to activate conversion of embedded PDF documents (PDF/A-3 only).

### 8.5 Changes in Version 4.10

- **Improved** conversion of transparent objects to PDF/A-1. E.g. filled paths that are transparent are converted to outlines in order to not cover underlying content when the transparency attribute is removed.
- **Improved** conversion of numbers that are larger than the implementation limit of PDF/A-1.
- **Improved** conversion of logical structure information (PDF/A level A).
- **Changed** behavior: Lock OCGs (layers) that need to be added to user interface (PDF/A-2 and PDF/A-3).

**Digital Signatures**

- **New** support for the new European PAdES norm (ETSI EN 319 142). See chapter “How to Create a PAdES Signature” in the user manual for more information.
- **New** support for the GlobalSign Digital Signing Service as cryptographic provider to create signatures and time-stamps.
- **New** signature algorithm RSA with SSA-PSS (PKCS#1v2.1) can be chosen by setting the provider session property `SigAlgo`.
- **New** conversion event 65536 when logical structure information is removed.
- **New** support for writing PDF objects into object streams. Most objects that are contained in object streams in the input document are now also stored in object streams in the output document. When enabling linearization, however, no objects are stored in object streams.
- **Improved** robustness against corrupt input PDF documents.
- **Improved** annotation appearance generation for polyline, squiggly, and stamp annotations.
- **Removed** the font ZapfDingbats.ttf from the product kit as it is not required anymore.
- **New** option `-st` to set signature format, e.g. for new European PAdES norm.
- **New** options `-afs1` and `-afs2` to set font size of the signature appearance.

### 8.6 Changes in Version 4.9

- **New** conversion features, e.g. improved conversion of TrueType font programs, ICC color profiles, or creation of annotation appearances.
- **New** supports for bold font simulation if only non-bold font is available in installed font directories.
- **Changed** behavior: The pre-analysis is now more strict, especially in certain corner cases of the PDF/A ISO Standard for which a conversion is strongly advised.
- **Improved** conversion of .notdef character for PDF/A-2 and PDF/A-3.
- **New** generated conversion event 4:
  - When text in input file is ambiguous.
  - When XFA (XML Forms Architecture) form data is removed.
  - When visual appearance of annotation cannot be created.
- **Improved** support for and robustness against corrupt input PDF documents.
- **Improved** repair of embedded font programs that are corrupt.
- **New** support for OpenType font collections in installed font collection.
- **Improved** metadata generation for standard PDF properties.

### 8.7 Changes in Version 4.8

- **New** conversion features, e.g. improved conversion of corrupt data such as fonts, text, or form XObjects.
- **New** conversion event 32768 when interactive elements such as actions or annotations are removed.
- **New** bold font simulation used when substituting bold with non-bold font.
- **Improved** creation of annotation appearances to use less memory and processing time.
- **Added** repair functionality for TrueType font programs whose glyphs are not ordered correctly.
9 Licensing, Copyright, and Contact

PDF Tools AG is a world leader in PDF (Portable Document Format) software, delivering reliable PDF products to international customers in all market segments.

PDF Tools AG provides server-based software products designed specifically for developers, integrators, consultants, customizing specialists and IT-departments. Thousands of companies worldwide use our products directly and hundreds of thousands of users benefit from the technology indirectly via a global network of OEM partners. The tools can be easily embedded into application programs and are available for a multitude of operating system platforms.

Licensing and Copyright  The 3-Heights™ PDF to PDF/A Converter Service is copyrighted. This user’s manual is also copyright protected; It may be copied and given away provided that it remains unchanged including the copyright notice.

Contact
PDF Tools AG
Kasernenstrasse 1
8184 Bachenbülach
Switzerland
http://www.pdf-tools.com
pdfsales@pdf-tools.com