

User Manual

# 3-Heights<sup>®</sup> Document Converter

Enterprise Edition

Version 6.27.14



# Contents

<b>1</b>	<b>Introduction</b>	<b>11</b>
1.1	Description	11
1.2	Product editions	12
1.2.1	Enterprise Edition	12
1.2.2	Small-Medium Enterprise Edition (SME)	13
1.3	Features	13
1.4	Applications	14
<b>2</b>	<b>Requirements</b>	<b>16</b>
2.1	Platform requirements	16
2.1.1	Complementary information	17
2.2	Operating systems	17
2.3	Office versions	17
<b>3</b>	<b>Installation</b>	<b>18</b>
3.1	Platform preparations	18
3.2	Service installation	18
3.2.1	Installation steps	18
3.3	Service configuration	20
3.3.1	License	20
3.3.2	Service configuration	20
3.3.3	Worker sessions	21
3.3.4	Worker office	21
3.3.5	Watched folder	22
3.3.6	Finish the service configuration	23
3.3.7	Configuration details	23
	Worker configuration details	24
	Office configuration details	24
	Outlook configuration details	24
3.4	Updating an existing installation	25
3.4.1	Uninstallation	26
<b>4</b>	<b>User guide</b>	<b>27</b>
4.1	Terms and abbreviations	27
4.2	Security considerations	27
4.3	Service configuration editor	28
4.3.1	General tab	28
4.3.2	Service configuration tab	28
	License key	29
	Additional workers	29
	Logging	29
4.3.3	Document conversion settings tab	29
4.4	Watched folders	30
4.4.1	Watched folders configuration	30
4.4.2	Control watched folder job	31
4.4.3	Implementation limits	32
4.5	Mail folders service	32
4.5.1	Mail folder configuration	32
	Requirements and limitations	33

4.6	Web service .....	33
4.6.1	Using MTOM with the web service .....	37
4.7	Client application .....	37
4.7.1	Features .....	38
4.7.2	Options .....	39
	Connection Mode .....	39
	Window Mode .....	39
	Output directory .....	40
	Override .....	40
	Selected profile .....	40
4.7.3	Explorer shell extension .....	40
	Configuration .....	41
4.7.4	Office add-in .....	41
4.8	Command line tool .....	41
<b>5</b>	<b>Processing guide .....</b>	<b>43</b>
5.1	PDF/A .....	43
5.2	Stamping .....	43
5.2.1	Pre-installed stamp files .....	43
5.2.2	Passing entire stamp files via API .....	44
5.2.3	References to fonts and images .....	44
5.3	Document transformations .....	44
5.4	Optimization .....	45
5.5	OCR .....	45
5.5.1	Use cases .....	46
	How to make text extractable .....	46
	How to detect and embed barcodes into metadata .....	47
5.6	Plugins .....	47
5.6.1	ScriptPlugin DLL .....	47
5.6.2	NetPlugin DLL .....	47
<b>6</b>	<b>Interface reference .....</b>	<b>49</b>
6.1	Document type index .....	49
6.2	Document extension index .....	49
6.3	Application options .....	50
6.3.1	Common to all external applications .....	51
	disabled .....	51
	Extensions .....	51
	MaxCallDuration .....	52
	preload .....	52
	RestartAfterConversions .....	52
	RestrictToExtension .....	52
	WorkingSet .....	52
6.3.2	[MSWord] INI-File Section .....	53
	BFFValidate .....	53
	BitmapMissingFonts .....	53
	CreateBookmarks .....	53
	DocStructureTags .....	53
	EmbeddedDocuments .....	53
	LockFields .....	54
	UpdateFieldsAtPrint .....	54
	PW .....	54

	ShowComments .....	54
	SAVEASPDF .....	55
	UseISO19005_1 .....	55
	VBAOFF .....	55
	Overriding settings .....	55
	PageHeight .....	56
	PaperMargins .....	56
	PageOrientation .....	56
	PaperSize .....	56
	PageWidth .....	56
	PRINTMARKUPS .....	56
	SoftErrFail .....	57
	LoadAllPages .....	57
6.3.3	[Excel] INI-File Section .....	57
	SAVEASPDF .....	57
	BFFValidate .....	57
	FailPageCount .....	58
	FitToPage .....	58
	ForceLetter .....	58
	MaxPages .....	58
	PaperSize .....	59
	Orientation .....	59
	PrintArea .....	59
	PW .....	59
	SHEET .....	60
	TIFF.DPI .....	60
6.3.4	[PowerPoint] INI-File Section .....	60
	BFFValidate .....	60
	BitmapMissingFonts .....	60
	DocStructureTags .....	61
	FitToPage .....	61
	OutputType .....	61
	PrintComments .....	61
	SAVEASPDF .....	61
	UseISO19005_1 .....	62
6.3.5	[Visio] INI-File Section .....	62
	SAVEASPDF .....	62
	SetActivePrinter .....	62
	USEEMF .....	62
	VisioPrintAll .....	63
6.3.6	[Outlook] INI-File Section .....	63
	access_grant .....	63
	button_yes .....	63
	OUTLOOKEXE .....	63
	SELECTFILES .....	63
	SKIPFILES .....	64
	UseHtmlBody .....	64
6.3.7	[MSProject] INI-File Section .....	64
	Chart .....	64
	DocStructureTags .....	64
	PercentScale .....	64
	PjPaperSize .....	65

	Portrait .....	65
	UseISO19005_1 .....	65
	SAVEASPDF .....	65
6.3.8	[00] INI-File Section .....	65
	SAVEASPDF .....	65
6.3.9	[Em1] INI-File Section .....	66
	HeaderTemplate .....	66
	Styles .....	66
	SELECTFILES .....	66
	SKIPFILES .....	66
	SkipUnusedInline .....	67
	DateFormat .....	67
6.3.10	[TXT2PDF] INI-File Section .....	67
	TEXT.BORDER .....	67
	TEXT.FONTNAME .....	68
	TEXT.FONTSIZE .....	68
	TEXT.HEIGHT .....	68
	TEXT.WIDTH .....	68
	TEXT.WRAP .....	68
	TEXT.LANG .....	68
	TEXT.TITLE .....	69
6.3.11	[EMF] INI-File Section .....	69
	AutoRotate .....	69
	Border .....	69
	PaperSize .....	69
6.3.12	[PDF] INI-File Section .....	69
	PDFA-PRECONVERT .....	69
	URLPerms .....	70
	RenderXFA .....	70
6.3.13	[Image] INI-File Section .....	70
	IMG2PDF.EXE .....	70
	FitImage .....	70
	Border .....	71
	CMRQUAL .....	71
	ResolutionDPI .....	71
	ThresholdDPI .....	71
	FailSize .....	71
6.3.14	[JPM] INI-File Section .....	72
	REPAIRSTREAMS .....	72
6.3.15	[XPS] INI-File Section .....	72
	MaxCallDuration .....	72
6.3.16	[PdfPrinter] INI-File Section .....	72
	Match .....	72
	PaperSize .....	72
	Resolution .....	73
	Quality .....	73
6.3.17	[TIFFPrinterBW] INI-File Section .....	73
	PaperSize .....	73
	Compression .....	73
	ResolutionX .....	73
	ResolutionY .....	73
	Quality .....	74

6.3.18	[TIFFPrinterColor] INI-File Section	74
	PaperSize	74
	Compression	74
	ResolutionX	74
	ResolutionY	74
	Quality	74
6.3.19	[HTML] INI-File Section	75
	IEPRINT	75
	HTMLMAXLOADTIME	75
	PageSetup	75
	HTMOPT	76
6.3.20	[ZIP] INI-File Section	76
	SKIPFILES	76
	SELECTFILES	76
	PROVIDER	77
6.3.21	[Job] INI-File Section	77
6.3.22	[PdfOptimize] INI-File Section	77
	Standard PDF optimization profiles	77
	custom	77
	exepath	77
6.3.23	[Options] INI-File Section	78
	AllowedTransforms	78
	MAXTRANSFORMTIME	78
	ForceClose	78
	plugins	78
	TerminateApps	78
6.3.24	[ACRORD] INI-File Section	79
	MaxCallDuration	79
6.3.25	[PDFOCR] INI-File Section	79
	MaxCallDuration	79
6.4	Job options	79
6.4.1	ALIVECHECK	79
6.4.2	CERTNAME	79
6.4.3	COLLCOMP	80
6.4.4	CONVERTALWAYS	80
6.4.5	EMBEDSOURCE	80
6.4.6	ERRSUMMARY	80
6.4.7	ERRPAGE	80
6.4.8	FlattenSignatures	80
6.4.9	FlattenFormFields	81
6.4.10	FORMAT	81
6.4.11	HTMLPRINTRESOLUTION	81
6.4.12	LINEARIZE	81
6.4.13	MAILHEADER	81
6.4.14	OCR	81
6.4.15	OCR.BITONAL	82
6.4.16	OCR.EMBEDBARCODES	82
6.4.17	OCR.ENGINE	82
6.4.18	OCR.LANGUAGE	82
6.4.19	OCR.IMAGEMODE	82
6.4.20	OCR.TEXTMODE	83
6.4.21	OCR.PAGEMODE	83

6.4.22	OCR.PARAMETERS	83
6.4.23	OCR.REEMBEDIIMAGE	84
6.4.24	OCR.ROTATEPAGE	84
6.4.25	OCR.TAGGING	84
6.4.26	PASSTHROUGH	85
6.4.27	PDFA	85
6.4.28	PDFA.ERROR	85
6.4.29	PDFA.LOGDETAILS	85
6.4.30	PDFA.LOGSUMMARY	85
6.4.31	PDFA.OCRMODE	85
6.4.32	PDFA.EMBEDALLFONTS	86
6.4.33	PDFA.SUBSET	86
6.4.34	PDFA.WARNCOLL	86
6.4.35	PDFA.WARNDOWNGRADE	86
6.4.36	PDFA.WARNUPGRADE	87
6.4.37	PDFA.WARNNOTPDFA	87
6.4.38	PDFA.WARNFONTSUBST	87
6.4.39	PDFA.WARNVISDIFF	87
6.4.40	PDFA.XMPWARNINGS	87
6.4.41	PDFA.CONVERTEMBPDF	87
6.4.42	PDF.COMPLIANCE	87
6.4.43	PDF.DATE	88
6.4.44	PDF.Embed	88
6.4.45	PDF.Info	88
6.4.46	PDF.InitialView	88
6.4.47	PDF.OWNERPASS	89
6.4.48	PDF.PERMISSION	89
6.4.49	PDF.Producer	90
6.4.50	PDF.USERPASS	90
6.4.51	PDFOPTIMIZE	90
6.4.52	SIGEMBEDOCSP	91
6.4.53	SIGFIELDS	91
6.4.54	SIGISSUER	91
6.4.55	SIGPAGE	91
6.4.56	SIGPROFILE	91
6.4.57	SIGPROVIDER	92
6.4.58	SIGPROXYCRED	92
6.4.59	SIGPROXYURL	92
6.4.60	SIGREASON	92
6.4.61	SIGRECT	92
6.4.62	SIGSTORE	92
6.4.63	SIGSTORELOCATION	93
6.4.64	SIGTSCRED	93
6.4.65	SIGTSURL	93
6.4.66	WARNSIGNATURES	93
6.4.67	STAMP	93
6.4.68	TEXTBODY	94
6.4.69	THUMBS	94
6.4.70	TIFF.BPI	94
6.4.71	TIFF.COMPR	94
6.4.72	TIFF.COMPR.BITON	94
6.4.73	TIFF.COMPR.CONTINUOUS	94

6.4.74	TIFF.COMPR.INDEXED	94
6.4.75	TIFF.COMPR.MRC	95
6.4.76	TIFF.COMPR.PDF	95
6.4.77	TIFF.DITHERINGMODE	95
6.4.78	TIFF.DPI	95
6.4.79	TIFF.RENDERINGMODE	95
6.4.80	TIFF.ROTATE	95
6.4.81	TIFF.UNPACK	96
6.4.82	TIFF.XDPI, TIFF.YDPI	96
6.4.83	ZIPPED	96
6.4.84	ZUGFeRD-invoice.xml:ADD	96
6.4.85	factur-x.xml:ADD	96
6.4.86	Special key	96
6.5	Document options	97
6.5.1	ADD	97
6.5.2	ATT.Name	97
6.5.3	ATT.Description	97
6.5.4	CMRQUAL	97
6.5.5	CODEPAGE	98
6.5.6	COLLCOMP	98
6.5.7	FAILFILES	98
6.5.8	FORCEAPP	98
6.5.9	HTZDOCS	99
6.5.10	Include Visio properties	99
6.5.11	MAILPARTS	99
6.5.12	ORIGINALNAME	99
6.5.13	AlternateText	100
6.5.14	LanguageTag	100
6.5.15	Outline	100
6.5.16	OutputType	100
6.5.17	PRINTCOLOR	100
6.5.18	PRINTDM	101
6.5.19	PW	101
6.5.20	SAVEASPDF	101
6.5.21	RemoveVBA	101
6.5.22	AcroRdEnableJS	102
6.5.23	TIFF.COMPR.BITONAL	102
6.5.24	TIFF.COMPR.CONTINUOUS	102
6.5.25	TIFF.COMPR.INDEXED	102
6.5.26	TIFF.COMPR.MRC	102
6.5.27	TIFF.UNPACK	102
6.5.28	TRANSFORM	103
6.6	Watched folder configuration	103
6.6.1	[O2PWFS] INI-File Section	103
	AutoDelete	103
	AutoDeleteAll	103
	JobPrefix	104
	KeepTimeForFailed	104
	KeepTimeForSucceeded	104
	LogLevel	104
	LogPath	104
	PollingInterval	104

	ServiceHost .....	105
	ServicePort .....	105
	Threads .....	105
	Thread .....	105
	WorkerThreads .....	105
6.7	Watched folder thread options .....	105
6.7.1	-b Set document options .....	105
6.7.2	-di Delete input files .....	106
6.7.3	-j Set converter job options .....	106
6.7.4	-l Create error log .....	106
6.7.5	-o Specify output directory .....	106
6.7.6	-of Specify failed directory .....	107
6.7.7	-op Unprefix output .....	107
6.7.8	-os Specify directory for succeeded jobs .....	107
6.7.9	-ow Ignore warnings .....	107
6.7.10	-owf Warnings output folder .....	107
6.7.11	-ox IDX name .....	108
6.7.12	-o0 Keep output .....	108
6.7.13	-o1 Second output directory .....	108
6.7.14	-o2 Force succeeded .....	108
6.7.15	-R Search for dropped files .....	108
6.7.16	-u Unzip output to folder .....	108
6.7.17	-w Specify the path to the root directory .....	109
6.7.18	-wd Specify the drop path .....	109
6.7.19	-wfi Ignore files with certain extensions .....	110
6.7.20	-wfs Process only files with certain extensions .....	110
6.7.21	-1 Single page output .....	110
6.7.22	-11 Single page output .....	110
6.8	Dispatcher service - O2PSRV.exe.config .....	110
6.9	Worker session control - O2PWSC.exe.config .....	114
6.10	Error codes .....	115
<b>7</b>	<b>Debugging .....</b>	<b>119</b>
7.1	Debugging of the dispatcher service .....	119
7.1.1	Debugging of worker sessions .....	119
7.2	Options .....	120
7.2.1	[Debug] INI-File Section .....	120
	DELAYEDCLOSE .....	120
	CLOSECRASHBOXES .....	120
7.2.2	Debug job option .....	120
	KeepTempFiles .....	120
<b>8</b>	<b>Troubleshooting .....</b>	<b>121</b>
<b>9</b>	<b>Version history .....</b>	<b>123</b>
9.1	Changes in versions 6.19–6.27 .....	123
9.2	Changes in versions 6.13–6.18 .....	123
9.3	Changes in versions 6.1–6.12 .....	123
9.4	Changes in version 5 .....	124
9.5	Changes in version 4.12 .....	124
9.6	Changes in version 4.11 .....	124
9.7	Changes in version 4.10 .....	125
9.8	Changes in version 4.9 .....	125

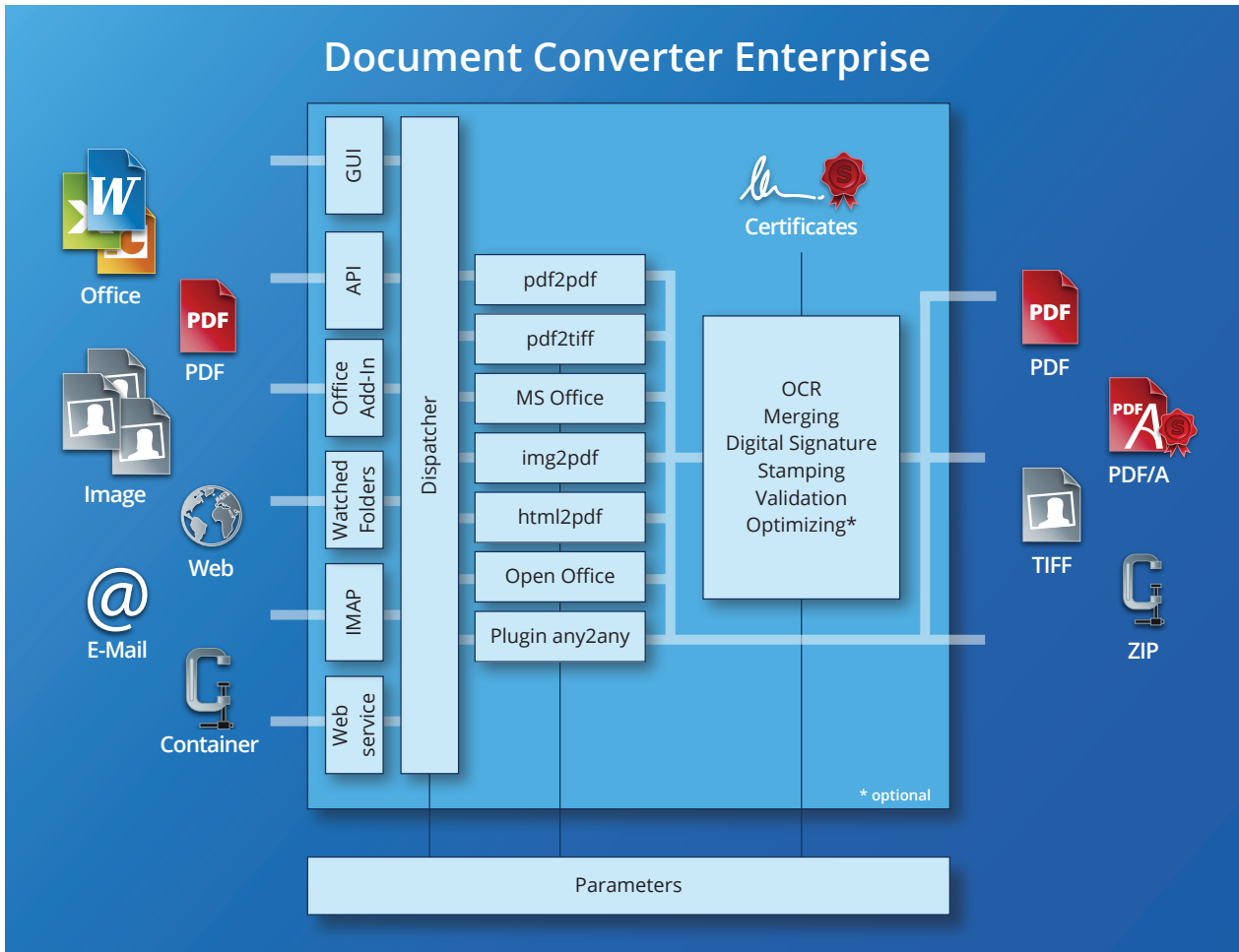
9.9	Changes in version 4.8 .....	125
<b>10</b>	<b>Licensing, copyright, and contact .....</b>	<b>127</b>
<b>A</b>	<b>Background information .....</b>	<b>128</b>
A.1	Configuration background information .....	128
A.1.1	TIFF output format .....	128
A.1.2	Conversion of Internet mail messages .....	129
A.2	Architecture background .....	130
A.2.1	Processing paradigm .....	130
A.2.2	Architecture .....	130
	Adhoc Converter Service .....	131
A.3	Windows policies and registry settings .....	131
A.3.1	Legal notice caption .....	131
A.3.2	Prompt for password .....	131
A.3.3	Security layer .....	132
A.3.4	Disabled allow list .....	132
A.3.5	Remote Desktop Connection settings .....	132
	MaxDisconnectionTime .....	132
	MaxConnectionTime .....	132
	MaxIdleTime .....	133
	RemoteAppLogoffTimeLimit .....	133
	fResetBroken .....	133
A.4	Installation parameters .....	133
A.5	Stamp file .....	134
A.5.1	Stamp file syntax .....	134
	Stamp .....	134
	Coordinates .....	137
	Modify content of existing stamps .....	137
	Stamp content .....	137
	Text .....	137
	Images and geometric shapes .....	141
	Transformations .....	142
A.5.2	Examples .....	143
	Example 1: Simple stamps .....	143
	Example 2: Modify "Simple Stamp" .....	143
	Example 3: Add watermark text diagonally across pages .....	144
	Example 4: Apply stamp to long edge of all pages .....	145
	Example 5: Stamp links .....	145
A.6	Samples .....	146
A.6.1	C .....	146
A.6.2	C# .NET .....	147
A.6.3	Visual Basic Script .....	147
A.6.4	Java .....	148
	Microsoft.NET based Java API .....	148
	Web service interface .....	148
A.6.5	Web service .....	150
	WSDL definition .....	150
	Web.config .....	154
A.7	Terminalserver .....	156
A.7.1	Platform requirements .....	156
A.7.2	Configuration of Terminal Services on Windows 2008 Server .....	157

A.7.3 Troubleshooting ..... 161  
A.8 Authorization of Exchange Online access ..... 161

# 1 Introduction

## 1.1 Description

The 3-Heights® Document Converter Enterprise Edition constitutes a solution for converting a wide range of document formats to PDF or TIFF. It can create PDF/A-1, PDF/A-2 and PDF/A-3 conforming files from office documents, images, or just simple text files.



The Document Converter supports various document formats and also includes support for unpacking and processing email attachments and ZIP or RAR archives that may be nested in arbitrary depth as well as various plugins for Microsoft applications.

### The main purpose of the Document Converter is to help you

- Make documents archivable (especially **MS Office** documents).
- Migrate existing archives.
- Exchange documents.
- Fulfilling standards (ISO) and security requirements.
- Archive websites and email.
- Standardize the variety of formats across the corporation.

## 1.2 Product editions

The 3-Heights® Document Converter comes in two different editions. Each edition is intended for a certain purpose. Contact the Pdftools sales team on the [Pdftools website](#) for concerns regarding the suitable 3-Heights® Document Converter edition.

### Quick Document Converter edition overview

		Enterprise Edition	SME Edition
Operation Systems	Service Components	Windows Server 2008 R2, 2012 or newer	Windows Server 2008 R2, 2012 Windows 7 or newer
	Client Applications	Windows 7 or newer	Windows 7 or newer
Formats	Office, Text	Yes	Yes
	PDF, Images	Yes	Yes
	Emails with Attachments	Yes	Yes
	Website	Yes	Yes
	Containers (e.g. ZIP)	Yes	Yes
	Custom (e.g. CAD)	Yes	
Features	Application/Interfaces	Watched Folders Explorer Shell Extension API Shell Watched Mail folders Web-Service	Watched Folders Explorer Shell Extension API Shell
	PDF/A-1, PDF/A-2, PDF/A-3	Yes	Yes
	TIFF Output	Yes	Yes
	Load-Balancing	Yes	
	OCR	Optional	Optional

### 1.2.1 Enterprise Edition

The Enterprise Edition of the 3-Heights® Document Converter not only offers the conversion of the common work documents, such as Office documents, but also the conversion of entire websites and emails. This makes it especially suitable for processing incoming mail, for archiving emails with attachments and Office documents in the long term. Additional formats can be added thanks to its plugin architecture. Moreover it is possible to execute a completion action after converting a document.

The Enterprise Edition features an API and a web-service, which allows a simple and quick integration in existing environments. Due to its high speed, availability and scalability it can be implemented as a ready-to-use solution in companies of any size.

There is a separate [3-Heights® Document Converter API documentation](#) available for the .NET interface.

## 1.2.2 Small-Medium Enterprise Edition (SME)

The Small-Medium Enterprise Edition offers all essential features that are required for the conversion of common Office documents to PDF/A or TIFF. Besides Office documents, all common image formats can be converted and there is an option to integrate an OCR engine as in the Enterprise Edition. The outstanding features of this edition are its uncomplicated installation and its applicability on Windows Clients.

The Small-Medium Enterprise Edition is a perfect solution for small and medium companies, which do not maintain their own IT department.

There is a separate installation and configuration documentation available on the Pdftools website for the [Small-Medium Enterprise Edition](#), which can be used as a quick-start.

## 1.3 Features

The 3-Heights® Document Converter combines the features of the 3-Heights® PDF Producer driver for Windows and the various document authoring applications such as Microsoft Word, Excel, PowerPoint, Visio, etc.

### The most significant features include

- Conversion of a variety of formats, such as Office documents or emails, to PDF/A and TIFF
- Merge documents
- Robust client/server architecture
- Quick and easy installation (MSI Installer)
- Configuration application for easy management of conversion settings
- Minimal platform and resources requirements for clients
- Convenient integration into user environment via explorer pop-up menu
- Automatic conversion processing from watched folders
- Apply digital signatures
- Scalability for high throughput by parallel processing
- Automatic processing also from folders on an email server
- Additional input document formats available via plugins
- Web service (IIS) interface
- Application Programming Interfaces available (C/C++, Java, COM)

### The supported document applications and input formats are

- Microsoft Office, including Word, Excel, PowerPoint, Visio, Outlook
- Document formats corresponding to the installed MS Office version
- MS Word accepts non-native formats, such as plain text (ASCII and UNICODE), WordPerfect, HTML and others
- PDF (Portable Document Format)
  - PDF 1.x (PDF 1.0, ..., PDF 1.7)
  - PDF 2.0
  - PDF/A-1a, PDF/A-1b

- PDF/A-2a, PDF/A-2u, PDF/A-2b
- PDF/A-3a, PDF/A-3u, PDF/A-3b
- EML und MSG (Internet Mail Message format)
- Image Documents (TIFF, JPEG, BMP, PNG, GIF, JBIG2, JP2, JPM, HEIC/HEIF)
- ZIP or RAR files containing documents corresponding to any supported format

## The supported output formats are

- PDF
  - PDF 1.x (PDF 1.0, ..., PDF 1.7)
  - PDF 2.0
  - PDF/A-1a, PDF/A-1b
  - PDF/A-2a, PDF/A-2b, PDF/A-2u
  - PDF/A-3a, PDF/A-3b, PDF/A-3u
- TIFF: Bi-tonal, Monochrome, Color; with CCITT G4, JPEG, flate, LZW or no compression

## Conformance

- 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 (ETSI TS 103 172) Part 2 and Part 4 (Long-Term Validation, LTV)
- Cryptographic Suites (ETSI TS 119 312)
- TIFF V6
- ZUGFeRD 1.0, Factur-X V1.0

## 1.4 Applications

The 3-Heights® Document Converter addresses applications that are in need of automated document conversion.

### Archiving

The 3-Heights® Document Converter is a great tool for archiving mixed collections of office documents, including email with attachments. It also accepts ZIP or RAR archives, and produces a single PDF/A-1, PDF/A-2 or PDF/A-3 conforming document containing the whole collection of input documents converted to PDF.

### Document assembly and publishing

The capability to convert and assemble documents can also serve to assemble multiple documents into one and prepare it for distribution.

### Integration in operation processes

To include the Document Converter in operation processes is supported in several ways. A convenient way to handle this, is via the "Watched Folders" service. Input documents just need to be copied into a dedicated folder to be automatically processed, with the output documents created in the configured output folder.

Even more flexibility is available by integrating the service into applications via the Document Converter API. Several interfaces such as a C, COM, and .NET are available.

The web service completes this list of features, it provides an additional interface of the Document Converter Enterprise Edition. This can be made accessible via the World Wide Web and is very easy to integrate with various technologies.

### **Interactive use**

The Document Converter accessible via interactive clients by several features, documents are converted on-the-fly. Typical documents are processed in only a few seconds to be presented to the client users.

There is no need to set up any infrastructure on the client side, making deployment as easy as possible.

# 2 Requirements

## 2.1 Platform requirements

The required hardware resources for a proper operation of the Document Converter depend on the type and amount of input documents and on the processing options used. Select progressively one of the following scenarios for a recommendation.

Of course the Hard Disk and Memory requirements are values which are used for the Document Converter only. For example, additional hard disk space is need to be considered for any Office applications to support the corresponding document formats.

**Note:** A straightforward way to verify if your platform meets the requirements for the Document Converter is that a session login and initialization should not take more than 20 - 30 seconds.

**For small Office documents; low throughput** (less than one page per second on average)

**Free hard disk space** Minimum 100 MB

**Available memory** Minimum 512 MB

**# CPU cores** Minimum 2

**For small Office documents; high throughput** (several pages per second on average)

**Free hard disk space** Minimum 200 MB

**Available memory** Minimum 1-2 GB

**# CPU cores** Minimum 4

**For complex documents**

**Free hard disk space** Minimum 200 MB

**Available memory** Minimum 1-2 GB

**# CPU cores** Minimum 2

**For raster image-based documents and conversion with OCR recognition; moderate throughput; OCR Service on separate computer**

**Free hard disk space** Minimum 200 MB

**Available memory** Minimum 512 MB

**# CPU cores** Minimum 2

**For an OCR service platform**

**Free hard disk space** Minimum 200 MB

**Available memory** Minimum 512 MB

**# CPU cores** Minimum 2

## 2.1.1 Complementary information

**Hard disk space** Is usually not an issue. The 3-Heights® Document Converter installation is small and disk space is determined mainly by space requirements for storing log and temporary files.

**Memory** Requirements are mostly determined by the type of conversions involved. Office applications may demand a significant amount of memory when source documents are large or complex.

Another reason for high memory demand can be high resolution conversion, e.g. PDF to TIFF, that is performed natively by the Document Converter. Using multiple worker sessions will also require more memory, as each session will occupy a certain amount of memory (which can vary from a few MB to several 100 MB depending on the type of documents being processed).

**# CPUs** The CPU load depends on one hand on the configured degree of concurrency (number of worker sessions) and the conversion load on the service. On the other hand, certain types of processing impose excessive CPU load by their nature, such as OCR processing and image (re-)compression.

As a rule of thumb, one extra CPU per additional worker session is sufficient, because most office applications will not make use of more than one CPU core in any circumstances.

## 2.2 Operating systems

The 3-Heights® Document Converter is available for the following operating systems:

- Windows Server 2008 R2, 2012, 2012 R2, 2016, 2019, 2022 | x64

## 2.3 Office versions

The 3-Heights® Document Converter is supported for use with the following Microsoft Office Versions and configurations:

- Microsoft Office 2016, 2019, 2021
- Office 365
- English, 64-bit version
- add language packs to support language-dependent features such as hyphenation

A product key (device license) must be installed and activated.

## 3 Installation

This chapter guides you through the installation of the 3-Heights® Document Converter Service. For information about Document Converter Client installation, see the Client manual.

For a successful installation it is important to meet certain platform and operating system requirements (see the previous chapter [Requirements](#)) and to prepare the platform accordingly.

### 3.1 Platform preparations

To prepare a platform for the installation of the 3-Heights® Document Converter, the following steps have to be done in advance:<sup>1</sup>

- 1. Install Windows OS**  
Install one of the supported Operating Systems and make sure the Spooler Service is enabled.
- 2. Install Microsoft .NET Framework 4.7**  
The Microsoft .NET Framework 4.7 or newer needs to be installed.
- 3. Install Universal CRT**  
The Universal C Runtime is installed by default on newer Windows platforms (Server 2016 and later). It is optionally distributed via Windows Update on older Windows versions.  
Verify that `ucrtbase.dll` is located in the Windows System32 folder. If it is missing, you may download and install the Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019 (`vc_redist*.exe`) from [download.microsoft.com](http://download.microsoft.com). We recommend to install both variants, x86 and x64.
- 4. Install Microsoft Office**
  - Select features according to the files that are intended to convert (i.e. Excel for `.xls` documents)
  - Install Service Packs
- 5. Install Adobe Reader 10 or higher** To enable the converter service to process PDF forms based on XFA that need rendering.
- 6. Install Internet Explorer** If the web site archiving feature shall be used or to convert HTML bodies of emails.

### 3.2 Service installation

Install the 3-Heights® Document Converter with assistance of the Microsoft Installer package

`DocumentConverterEnterprise-<version>-Windows-(<platform>).msi`

, where `<version>` is the version number and `<platform>` is either `32bit` or `64bit`. The 32-bit version runs on both, 32 and 64-bit platforms. The 64-bit version runs on 64-bit platforms only. Install only one version, on 64-bit systems preferably the 64-bit version.

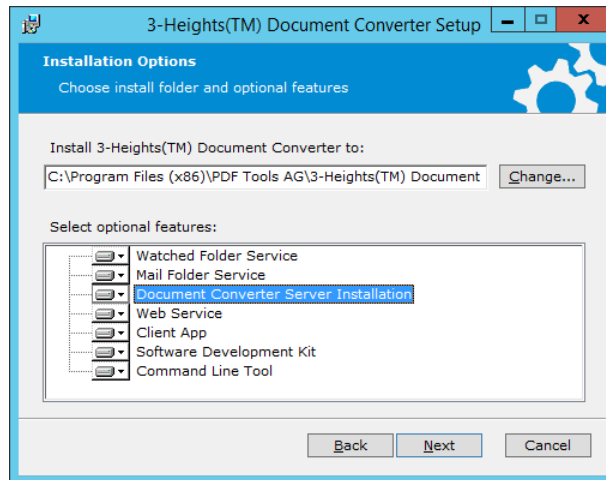
Moreover, public properties of the installer package can be specified, see [Installation parameters](#).

#### 3.2.1 Installation steps

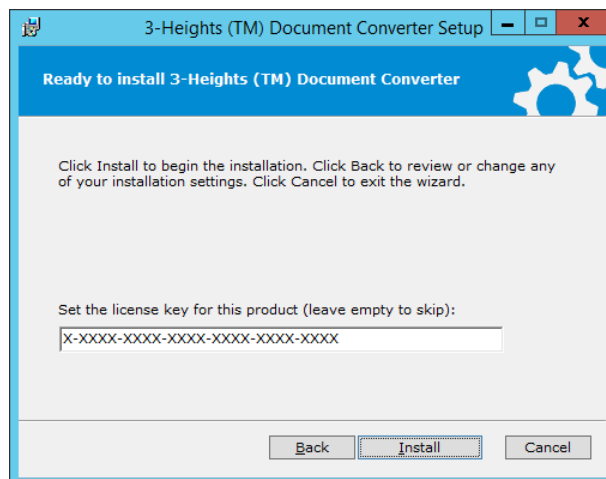
- Run the `DocumentConverterEnterprise-<version>-Windows-(<platform>).msi`.
- After going through the initial setup page, accept the terms in the "License Agreement".
- Continue to the Installation Options via the "Next" button.

<sup>1</sup> If the 3-Heights® Document Converter is used in a "Terminal Server Environment" see [Configuration of Terminal Services on Windows 2008 Server](#)

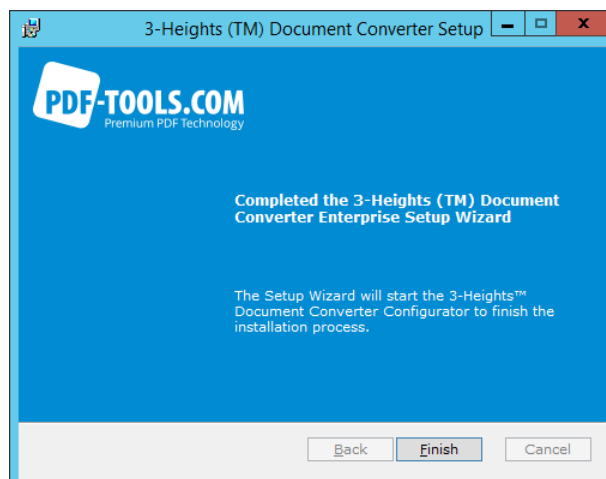
- a. Change the destination folder, if desired.
- b. Select the features to be installed.
- c. Click on "Next".



4. The license key can be specified before the installation process.



5. Trigger the installation by clicking on the "Install" button.
6. After the installation is completed, click on the "Finish" button to close the Setup Wizard and start the Document Converter Configurator.

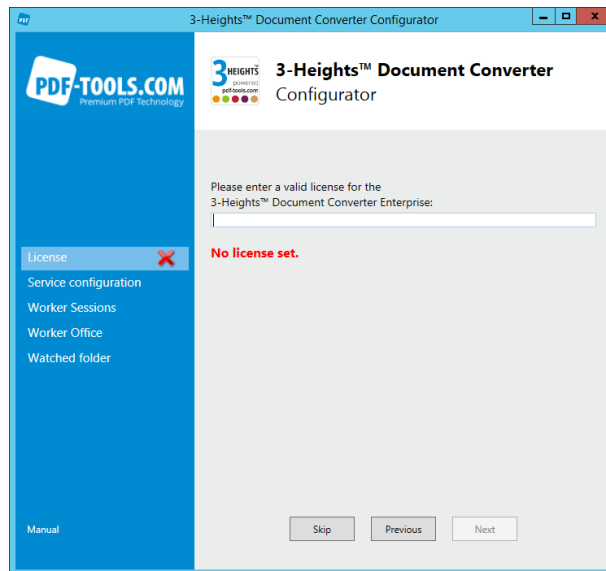


## 3.3 Service configuration

The 3-Heights® Document Converter Configurator is started automatically when the Document Converter was installed for the first time. It eases and guides through the Service Configuration process.

### 3.3.1 License

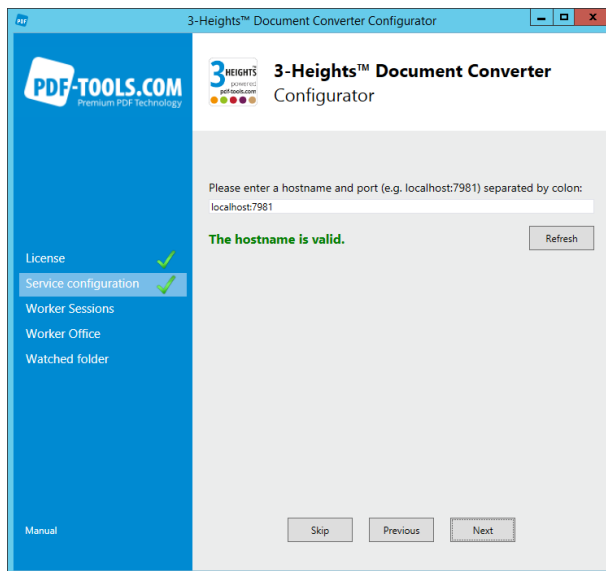
A valid license for the 3-Heights® Document Converter Enterprise Edition is required. To maintain multiple licenses please use the 3-Heights® License Manager that was installed with the server components.



### 3.3.2 Service configuration

3-Heights® Document Converter Clients can access the Document Converter Service if they are located in the same network.

Therefore the service needs a hostname that is available for the clients, `localhost` can be used if both installations are on the same machine. If the access is through a network connection, the firewalls on the route between the service and the client machines must be configured to permit inbound connections on the configured port (default is 7981).

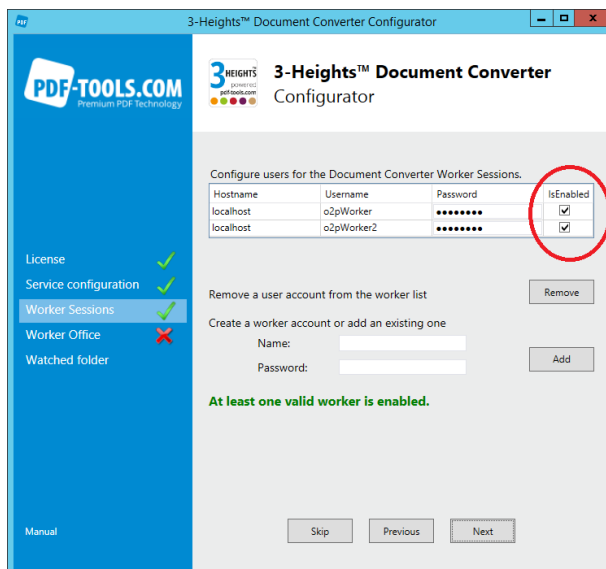


### 3.3.3 Worker sessions

For a full fletched use of the Document Converter Enterprise Edition it is necessary to configure Worker Accounts. On the page “Worker Sessions” you can create new accounts or use an existing one as Worker Account. If the server is a domain member, make sure to fully qualify the user name (e.g. CONVSrv\o2p).

It is important that for each Worker Session there is a separate Worker Account and moreover that every Worker Account is not used for other purposes. A Worker Account is used by the Document Converter if it is on the Worker list and marked as “IsEnabled”.

**Note:** Worker Accounts are user or domain accounts which cannot be deleted with the Configurator.



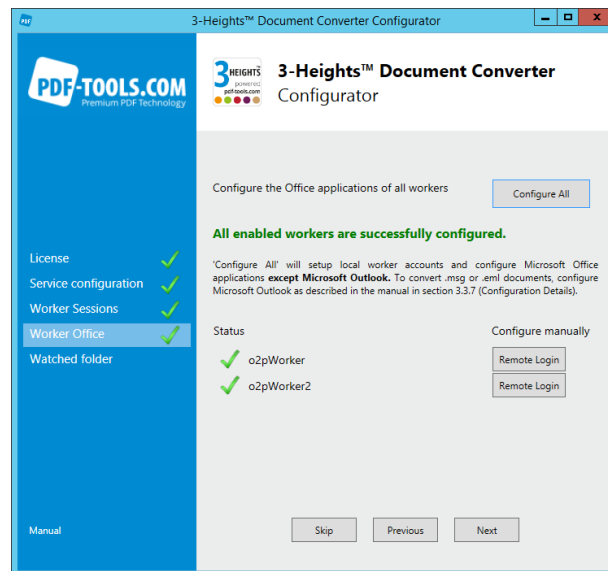
### 3.3.4 Worker office

After sufficiently Worker Accounts are added to the Document Converter configuration, for every Worker Account the Office applications need to be configured too. Therefore click the button “Configure All” on the “Worker Office”

page of the Configurator. This starts for every Worker Account the configuration process separately and takes up to 90 seconds for every worker.

If the Office configuration fails for certain Worker Accounts it is possible to do this manually via the “Remote Login” button next to the corresponding Worker Account name. This starts an interactive remote desktop session in a new window. In which the user can configure the Office applications according to the section [Office configuration details](#).

**Note:** The configuration status does not change after a manual configuration of the Office applications.



If you want to use Microsoft Outlook to convert .msg documents, Microsoft Outlook needs to be configured manually in any case via a remote desktop session and as described in section [Outlook configuration details](#).

**Note:** The default value in this Document Converter version is NOT to use Outlook to convert .msg.

### 3.3.5 Watched folder

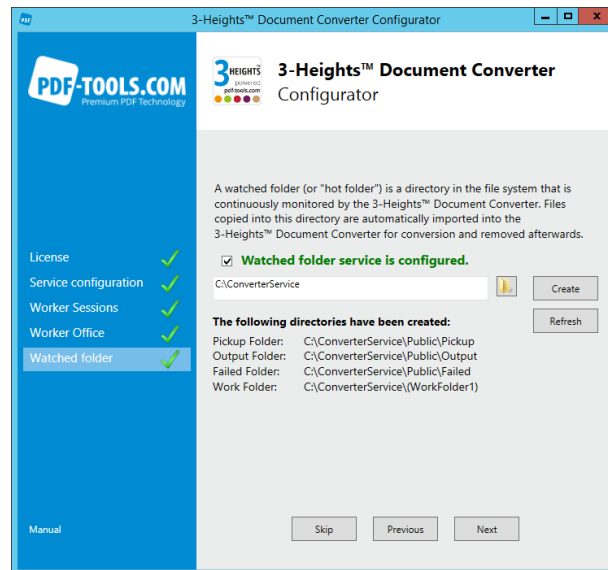
The directories of a Watched Folders Service can be specified and created with the Configurator on the next page.

1. Mark the corresponding checkbox.
2. Set an existing root directory.
3. Click on the button “Create” to automatically create the directories as shown in the preview:
  - an empty Work Folder
  - the folder Public with the subfolders Failed, Output and Pickup

**Note:** It is highly recommended to use **local** folders for the watched folder service, network folders can be temporary unavailable and disturb the conversion process.

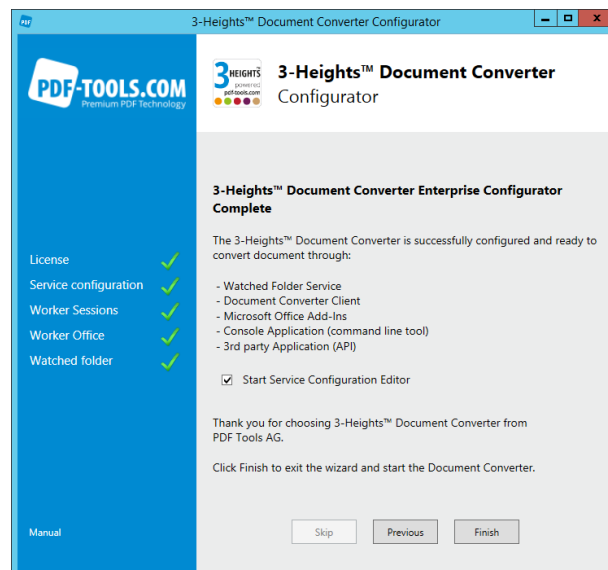
The watched folder service is installed using the LOCAL SYSTEM account. Ensure that this user has read and write access to all folders (Pickup, Output, Failed, Work).

For information regarding the usage see section [Watched folders](#) and regarding the configuration see section [Watched folder configuration](#).



### 3.3.6 Finish the service configuration

To run a reasonable Document Converter Service the last page of the Configurator should look like on the screenshot below:



- Mark the checkbox "Start Service Configuration Editor" to start the Document Converter Editor to configure the conversion process afterwards. See section [Service configuration editor](#) for guidance.
- Click on the button "Finish" to close the Configurator and start Document Converter service.

### 3.3.7 Configuration details

This section provides supplementary details for configuration tasks not covered by the 3-Heights® Document Converter Configurator.

## Worker configuration details

**Create worker accounts** Configure at least one worker account for the Document Converter. A worker account needs either a local Windows user account or an domain user account.

It is strongly recommended to use a separate, dedicated Windows user accounts for each worker session. Depending on CPU and memory resources available, reasonable numbers of sessions are two to four.

**Enable remote desktop** Under "Control Panel" → "System and Security" → "System" → "Remote Settings" → "Remote" Tab verify that Remote Desktop connections on the server are enabled.

On the same Tab click on "Select Users..." and add all service accounts which are used for worker sessions.

**Log in** For each Windows Account that is linked to a Worker account, log in via Remote Desktop (`mstsc.exe`) on the server, where the installation took place. Start each of the MS Office applications once.

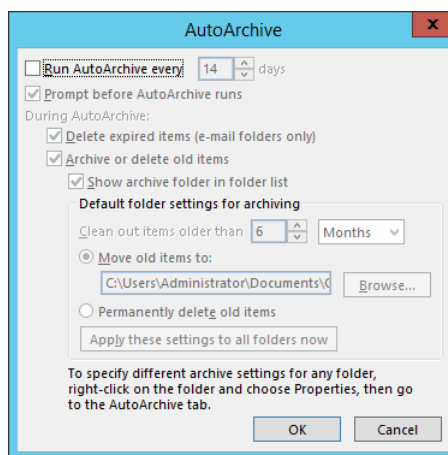
Make sure to open a document of each format you need support for (e.g. a .WPD document from Word, a SVG document from Visio, etc.). This procedure is required to complete the Office installation and to verify its completeness.

## Office configuration details

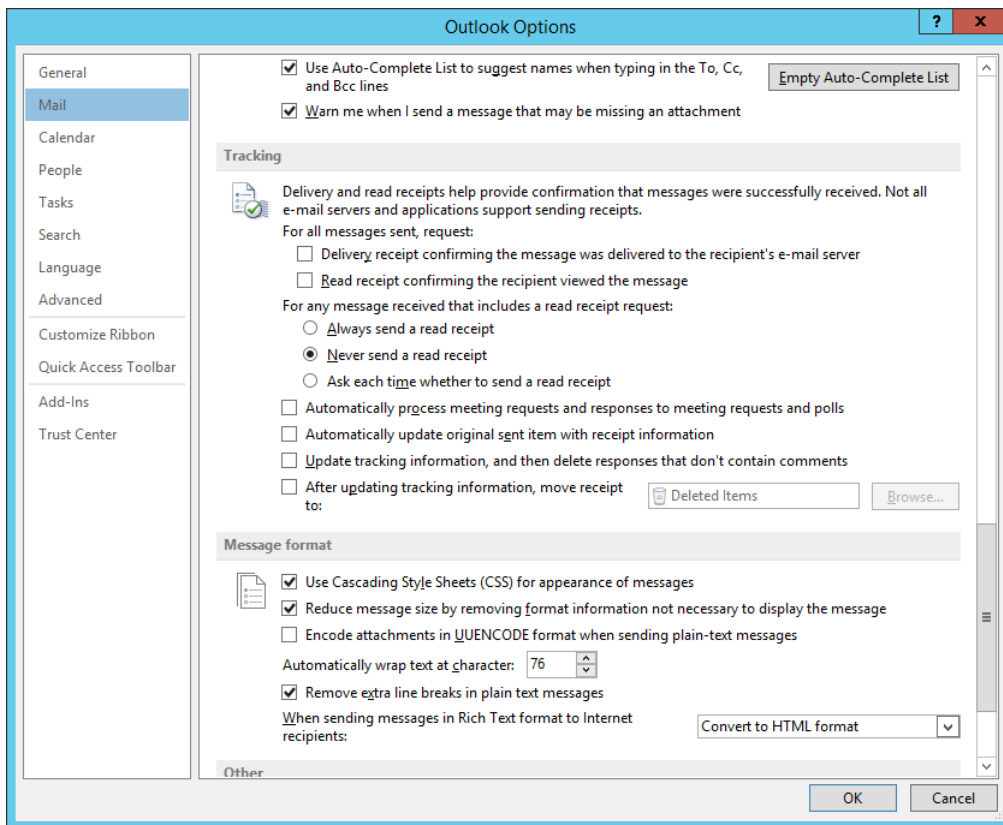
1. Activate the license of the Office application.
2. Set the format options in the Office application.
3. Open a sample document for each format that should be converted with the Document Converter. Ensure that there are no pop-up dialog boxes that must be closed manually.

## Outlook configuration details

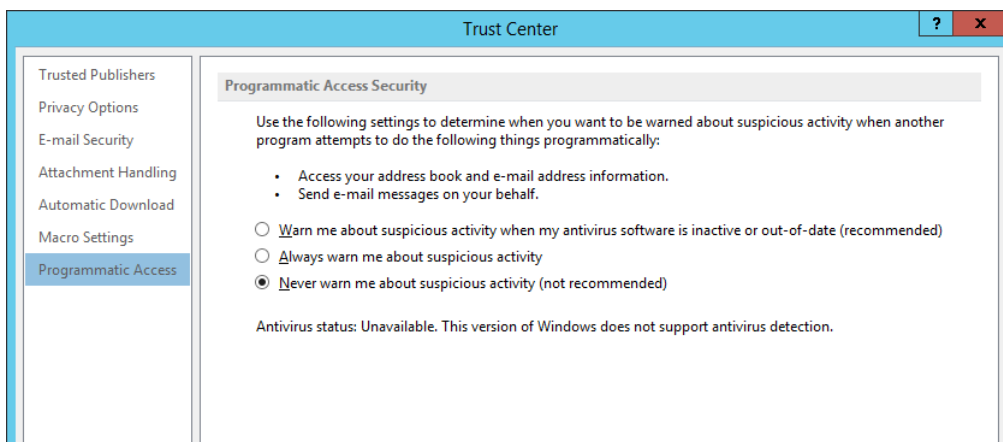
- Ensure that there are no pop-up dialog boxes that must be closed manually.
- Disable automatic features, such as "AutoArchive".  
"Options" → "Advanced" → "AutoArchive" → "AutoArchiveSettings..."



- Disable automatic printing of mail attachments.
- Disable reception confirmations or prompting for these.



- Disable notifications for programmatic access.  
as Administrator: "Options" → "Trust Center Settings" → "Programmatic Access" → "Never warn..."



A specific user can start Outlook only once on a server, because it exclusively locks the user's Outlook.pst. Thus, a different user account must be specified for each worker session.

## 3.4 Updating an existing installation

To update a 3-Heights® Document Converter Enterprise Edition version

1. Stop the Document Converter Service (all services).
2. Make sure no other users are logged into the server. Force any other sessions to terminate.
3. Make sure the Print Spooler is (still) running.

4. Terminate any office applications or other programs having a “print” function and might have loaded a printer driver DLL. In case you have created a share for one of the PDF or TIFF printers, remove that share. Otherwise, uninstallation of the printer drivers will fail.
5. Execute the MSI of the newest release and follow the steps as described in the section [Service installation](#). If this yells an error, an uninstallation must be performed before installing the newest Document Converter Enterprise Edition release.

### 3.4.1 Uninstallation

1. Stop the Document Converter Service (all services).
2. Stop the Web Service if installed and remove from IIS.
3. Make sure no other users are logged into the server. Force any other sessions to terminate.
4. Restart the Print Spooler service (at least, be sure this service is running).
5. Terminate any office applications or other programs having a “print” function and might have loaded a printer driver DLL. In case you have created a share for one of the PDF or TIFF printers, remove that share. Otherwise, uninstallation of the printer drivers will fail.
6. Perform the uninstallation (from the “programs and features” control panel or using the MSI kit).

**Note:** The uninstallation does not remove the configuration data contained in \*.config, \*.xml and \*.ini files. These files remain in the installation directory, and will not be changed when performing the installation of the new version into the same installation directory.

## 4 User guide

This section gives an overview of the usage and configuration for the 3-Heights® Document Converter Enterprise Edition standard features.

### 4.1 Terms and abbreviations

The described terms below are important for the comprehensibility of the Document Converter.

**Application options** These options control the corresponding application (e.g. MS Word). Application options are set with the 3-Heights® Document Converter Service Configuration Editor on the “Document Conversion Settings” tab or directly in the `O2PWSC.ini` file.

**Job** This term stands for a conversion job, which is the common paradigm for the Watched Folder Service (`O2PWFS.exe`) and the Command Line Tool (`o2pclient.exe`).

The two executables actually build on top of the API DLL `O2PProxyAPI.dll`, for detailed information one is referred to the section “Job Structure” in the [3-Heights® Document Converter API documentation](#).

**Job options** Are used to configure the general conversion process. Job options can be set analogous to application options or can be specified individually for the different features.

**Note:** If job options are set at different points, within the conversion process, all job options are merged together. In case of conflicts, those job options which are set for the feature in use, override those of the Service Configuration.

All available document options can be set as job options, but not vice versa. Moreover, settings specified at the document level have priority over settings specified at the job level.

**Document options** These options are set to process a particular document and are set “on the job” (i.e. set individually for each feature). The distinction between document options in job options is due to different methods in the `O2PProxyAPI.dll`.

**Note:** Certain document options can control an application also (i.e. override application options).

### 4.2 Security considerations

The Document Converter is not designed for use in a hostile environment such as the Internet.

The service port should be protected by a firewall to protect it from unauthorized use or malicious attacks. Client components accessing the service need not to provide any credentials. Make sure to install these components only on computers on which users are permitted to use them.

Log directories and files on the computer hosting the server components are not secured. You may want to restrict access from unauthorized users by tailoring the security settings of the `log` and `Temp` folders.

**Note:** Write access must be granted to the following accounts:

- The account configured for the dispatcher service (O2PSRV).
- The account(s) configured for the worker sessions

Similar considerations apply to the directories used by the Watched Folder Service, if installed.

## 4.3 Service configuration editor

The document converting configurations are conveniently set with the 3-Heights® Document Converter Service Configuration Editor (O2PConfigure.exe). This editor is an intuitive tool to set a very broad range of options.

To set and maintain general service configurations use the first three tabs of the editor.

### 4.3.1 General tab

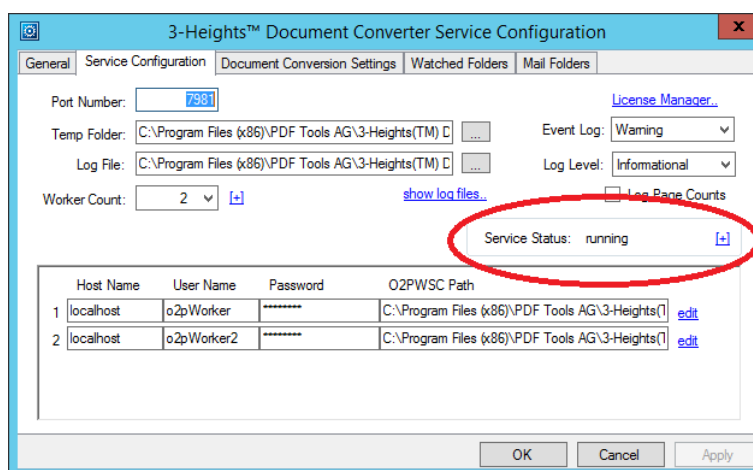
The first tab on the Configuration Editor provides standard information about the 3-Heights® Document Converter Service, such as the Document Converter version, status and the path to its installation directory.

### 4.3.2 Service configuration tab

This tab controls the Document Converter Service.

**Important:** After changes are made, click the “Apply” button and restart the service

**Hint:** Hold the “Ctrl” key and right-click on this tab to display the number of processed document conversions.



**Start and stop the service** Start and stop the service with the [ + ] next to “Service Status”.

If the service is started in “(debug)” mode, the Dispatcher connects visible one by one to all configured workers via a remote desktop connection.

### [ + ] Adjust printers

**Remove all entries** Remove all printers associated to the Document Converter Workers.

**create PDF and TIFF printers** Create an accurate number of new printers.

## License key

The message "License key missing" is printed in red in the upper right corner of this tab, if there was no valid license key specified. Open the "License Manager.." and enter a valid license key.

## Additional workers

**Note:** The simplest way to add more Worker Accounts is to run the Document Converter Configurator again and configure those as described in the sections [Worker sessions](#) and [Worker office](#).

Nevertheless it is possible to add (not create) Worker Accounts directly on the "Service Configuration" tab:

1. Increase the "Worker Count" on the "Service Configuration" tab.
2. Update the "User Name" and "Password" fields according to the settings of the service accounts that shall be used by the worker sessions. If the server is a domain member, make sure to fully qualify the user name (e.g. CONVSRV\o2p).
3. Run the service in "debug" mode to log in the accounts automatically.
4. When increasing the number of workers, the number of PDF Producer printer entries may need to be adjusted, see [\[+\] Adjust printers](#). The Configuration Editor will automatically attempt to do so upon saving the updated settings.

If this adjustment fails, a message box saying "The PDF Producer configuration update failed" will pop up; the settings have been saved, but the PDF Producer printer entries for any additional worker sessions may be missing. If you encounter this problem, restore the previous worker count and contact Pdftools support.

## Logging

- Set a level for the "Event Log" and "Log Level". Common log settings are "Warning" or "Error".
- Logging of page counts can be enabled by checking the "Log Page Counts" box on the Service Configuration tab. An entry consisting of date time and number of pages is appended for each conversion job.
- Additional, accounting relevant information can be passed through to the log file from the job options (any job level key/value pair setting with a key prefix of ACCOUNTING . will be passed).

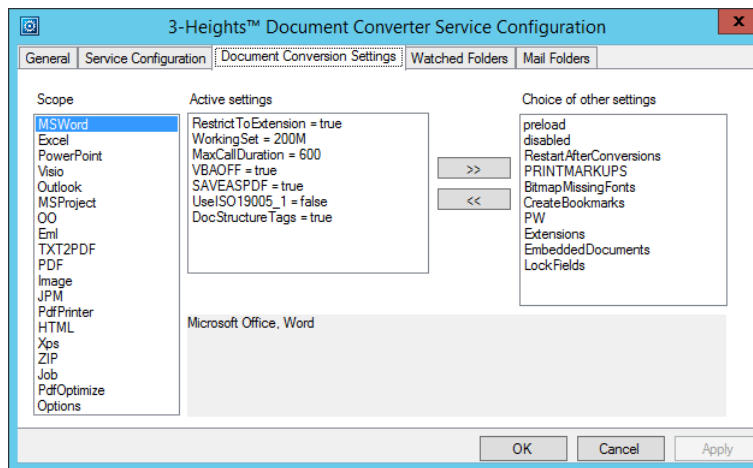
The name of these log files consists of the prefix `ac` followed by the current year and month.

The log information is stored in text files that are located in the Document Converter's log directory.

### 4.3.3 Document conversion settings tab

Adjust on this tab the service document conversion settings according to specific needs. Therefore select first the "Scope" and then the according settings. Common settings are already listed in the editor. See the section [Application options](#) as well as [Job options](#) and [Document options](#) for a description of the available options. Moreover, consider the section [Terms and abbreviations](#) to comprehend the hierarchy of options set on this tab or at a different point in the conversion process.

The conversion settings are saved in the `O2PWSC.ini` file.



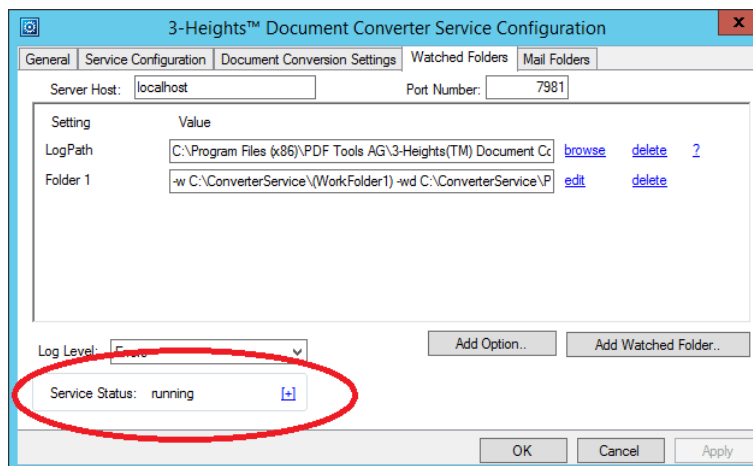
## 4.4 Watched folders

Once you have a Watched Folders Service configured, it is very simple and straight-forward to use. Documents can simply be moved into the pickup folder, and later the conversion results can be retrieved from the output folder.

**Note:** The input file will be moved to the `Failed` folder in case of a failed conversion. An output file may still be stored in case of non-fatal errors (such as PDF/A conversion issues).

Use the “Service Configuration Editor” to set up and control the Watched Folders Service. The corresponding INI file `O2PWFS.ini` is located in the installation directory.

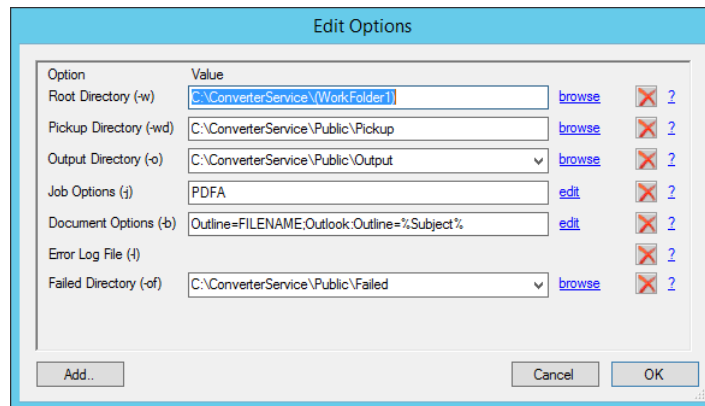
**Start watched folders service** This Service needs to be started and stopped separately with the `[+]`, see screenshot below.



### 4.4.1 Watched folders configuration

**“Add option..”** Set the options [AutoDelete](#), [AutoDeleteAll](#) and [JobPrefix](#) directly on the “Watched Folders” tab. In order to move input files rather than delete them, [AutoDelete](#) and [AutoDeleteAll](#) settings must be set to **false**.

**edit "Folder #"** Enter the specific options directly into the value field, or open the pop-up box by clicking on "edit". Directories can be specified as an absolute path or prefixed by a colon as a subfolder of the pickup directory. For a detailed description of the watched folder settings see the section [Watched folder thread options](#).



### The service creates the following subfolders within the specified root directory

**Failed** Dropped files are moved to this folder when a failure is encountered.

**InProgress** Dropped files are moved to this folder while being processed.

**Jobs** Dropped files are moved here; they are renamed to contain a prefix consisting of date and time, so they will be processed in the order they were dropped.

**Logs** Log files for files that failed during processing are stored here; the log file name consists of the original file's name and the extension `.txt`.

**PDFs/TIFFs** This directory contains the PDF/TIFF output of successfully processed files.

**Note:** The name of this directory depends on the [FORMAT](#) setting in the job options specified for the folder (unless option `-o` is specified).

**Succeeded** This directory contains dropped input files that have been successfully processed (when `AutoDelete=false` is specified in the INI file).

## 4.4.2 Control watched folder job

The Watched Folder Service accepts [Job options](#) and [Document options](#) that are specified in the corresponding [Thread](#) declaration in the `O2PWFS.ini` file (see section [Watched folder configuration](#)), or in a job control file.

**A job control file** is a text file that is dropped into the watched folder, having the file extension `.control`. Each line of the text file either consists of an option setting, or a document file path. The documents referenced in the control file are left untouched.

**Note:** Each thread uses a single converter job; since job options are reset before beginning processing of a new conversion, settings in a control file will be confined to the scope of the control file.

**Example:** Job control file

```
-b Outline=FILENAME;Outlook:Outline=%Subject%  
\\ComputerName\ShareName\document1.rtf  
\\ComputerName\ShareName\document2.xls  
\\ComputerName\ShareName\document3.msg
```

The watched folder service will move the control file to the succeeded folder (or delete it) on job completion.

### 4.4.3 Implementation limits

The Watched Folder Service has the following implementation limits:

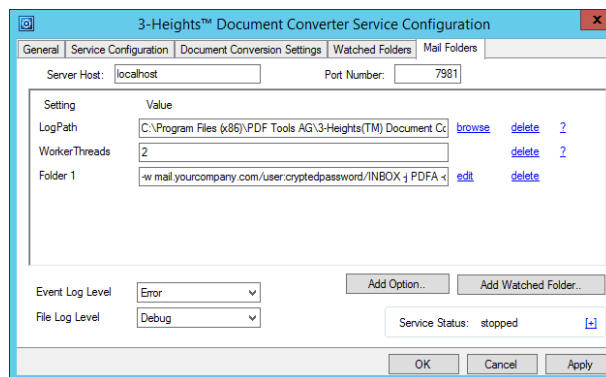
**Maximum full path name length for input documents** Limited by Windows (260 characters).

**Maximum full path name length of output documents** Limited to 160 characters, unless option `-u` is specified. With `-u`, there is a limit of 80 characters for the output folder, and 80 characters for the names of files stored in that folder (file extensions included; the significant part of the names will be less).

**Maximum number of watched folders** There is no hard limit for the maximum number of watched folders; however, file search overhead will increase with the number of watched folders. The number may exceed a few hundred for folders on a local disk, but should be significantly less for network shares.

Processing of documents from a specific folder is performed sequentially. If you need to increase throughput, create multiple folders and distribute documents equally.

## 4.5 Mail folders service



When selected during MSI based installation, the Mail Folders service "O2PMFS" is installed. This service features the conversion of email messages to PDF/A or TIFF – directly from mail folders on a mail server.

Use the "Service Configuration Editor" (O2Pconfigure.exe) to set up and control the Mail Folders Service. The corresponding INI file o2pmfs.ini is located in the installation directory. Its syntax and options are similar to the O2PWFS.ini file.

**Start mail folders service** This Service needs to be started and stopped separately with the [+], analogue to the Watched Folders Service.

### 4.5.1 Mail folder configuration

**edit "Folder #"** Select the "Mail Folders" tab on the "Configuration Editor" and enter either the specific options directly into the value field, or open the pop-up box by clicking on "edit".

Enter all necessary mail credentials to login the desired mail account for the "Pickup", "Output", "Failed" and "Succeeded Folder". Each folder can be specified as type "File System Directory" also.

**Note:** If the **Output Folder** is of type "File System Directory" it is possible to specify a "Completion Action", i.e. you can choose an executable which is called when a document is processed in this directory.

To configure non-standard ports for IMAP or SMTP connections as well as SSL/TLS, append a colon followed by the port number to the server name. For SSL, add a small letter s; to configure SSL with STARTTLS, add a capital letter S.

To configure Microsoft's Live Mail server for example, specify "smtp.live.com:587S" (smtp.live.com is the network name, 587 the port number, and S indicates that "StartTLS" shall be used).

**Note:** Passwords for mail accounts are encrypted in the configuration file.

#### "Add option.."

**FileNameTemplate** Use this option to construct the file name from email contents. Possible place holders are %SUBJ, %DATE, %FROM, %TO and %GUID.

**MaxAttachmentNameLength** (default: 240 characters) Maximum length for naming mail attachments containing conversion results; the name is derived from the value of **FileNameTemplate**.

**WorkerThreads** Assign the number of worker threads to the Mail Folders Service.

## Requirements and limitations

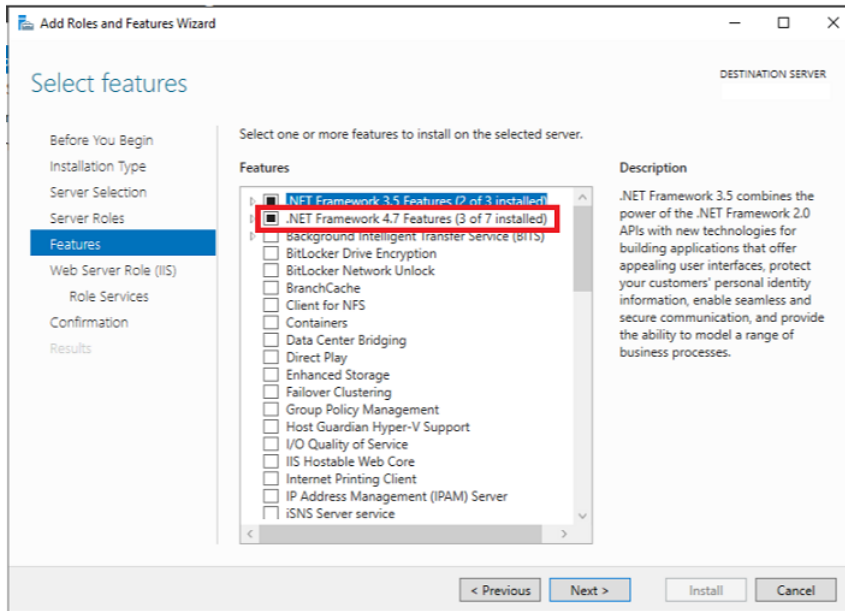
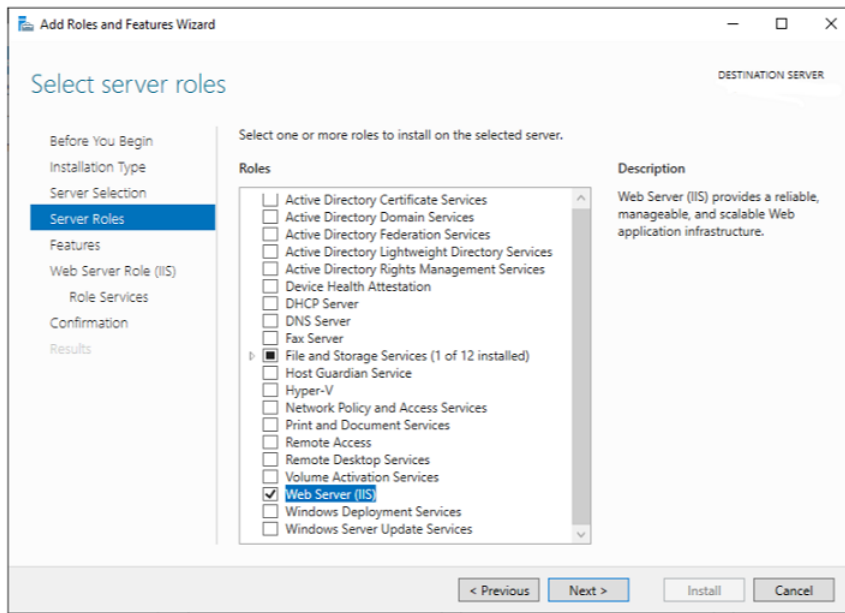
The only protocol to access a mail server directly is via IMAP. SSL is supported. For authentication, plain text and OAuth2 with Microsoft Exchange Online is supported. See appendix [Authorization of Exchange Online access](#) for information about configuring the Exchange Online client application.

As IMAP does not support locking, make sure that folders watched by the service are not simultaneously accessed and modified by users or another instance of the service.

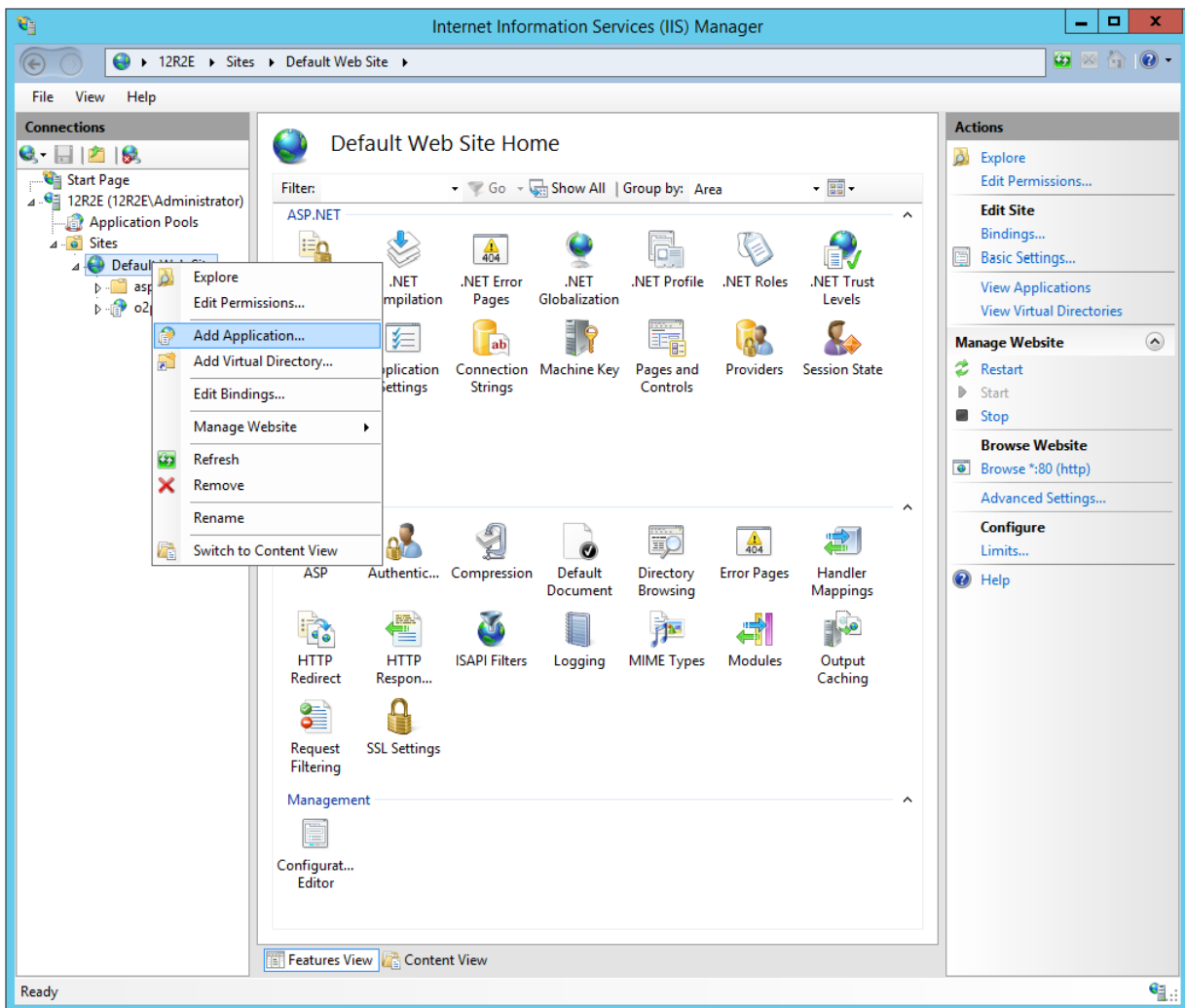
## 4.6 Web service

A web service component provides document conversion services to SOAP based client applications. When this feature is selected for installation, the installation directory will contain a folder named **WebService**.

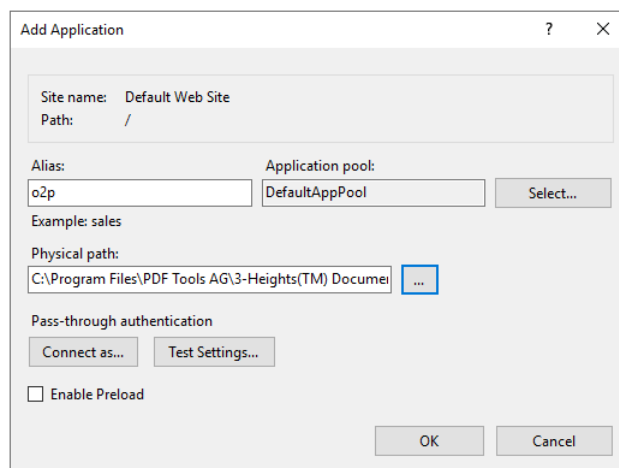
The post-installation procedure for the web service consists of adding the "WebService" directory as a virtual folder to an Internet Information Services (IIS), and defining this folder as an application. Make sure ASP.NET is enabled, and Microsoft.NET V4.7 is configured. The following screen shots illustrate these steps on a Windows Server 2019 platform with IIS 10.



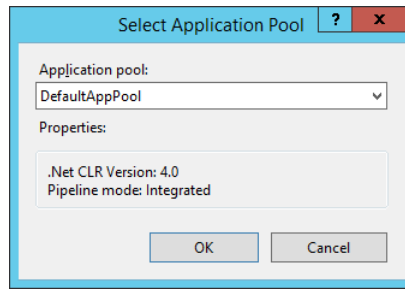
Make sure Internet Information Services (IIS) is installed with the necessary features (ASP.NET 4.7).



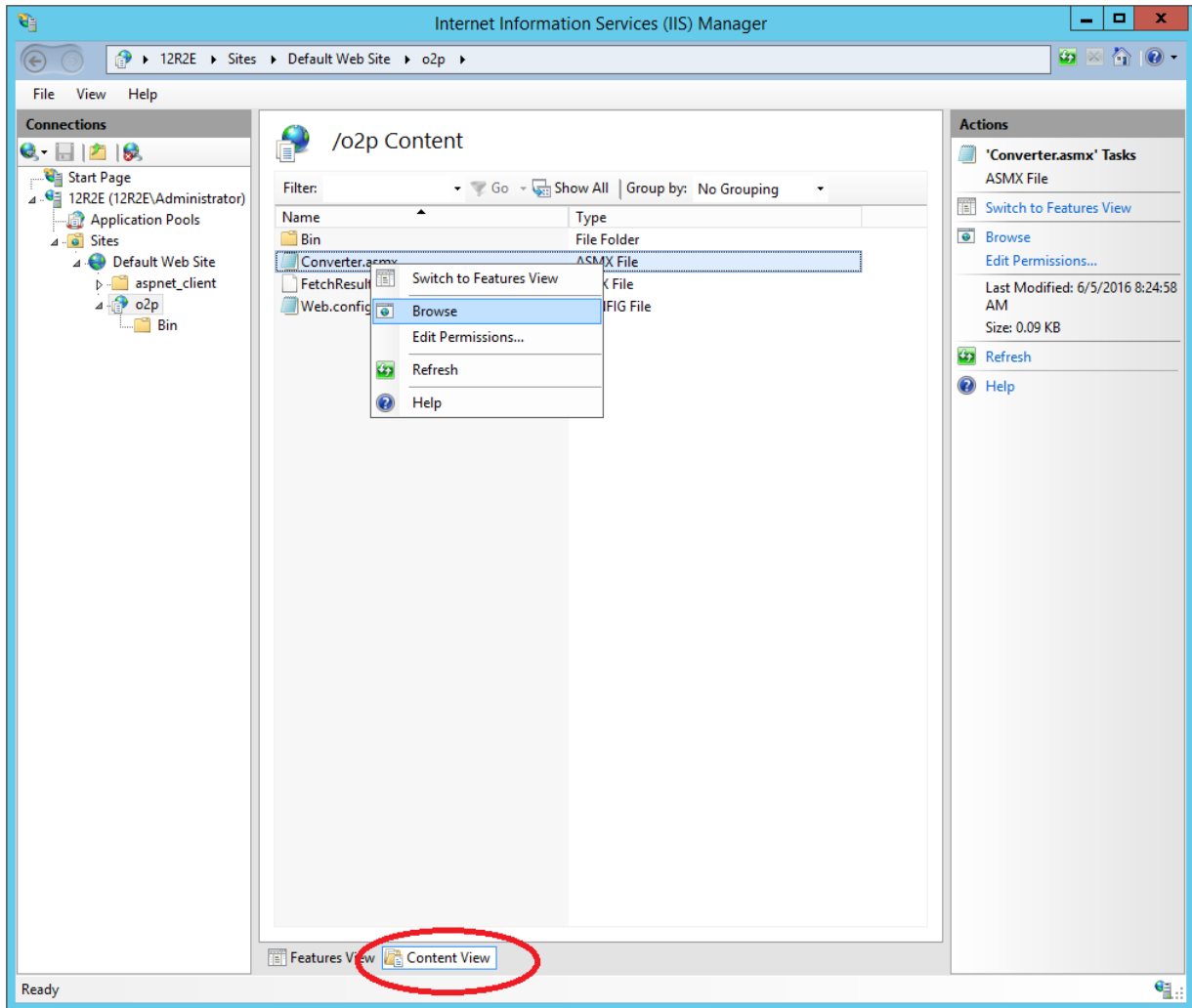
Open “Internet Information Services (IIS) Manager” from the “Administrative Tools” Menu; then use “Add Application”.



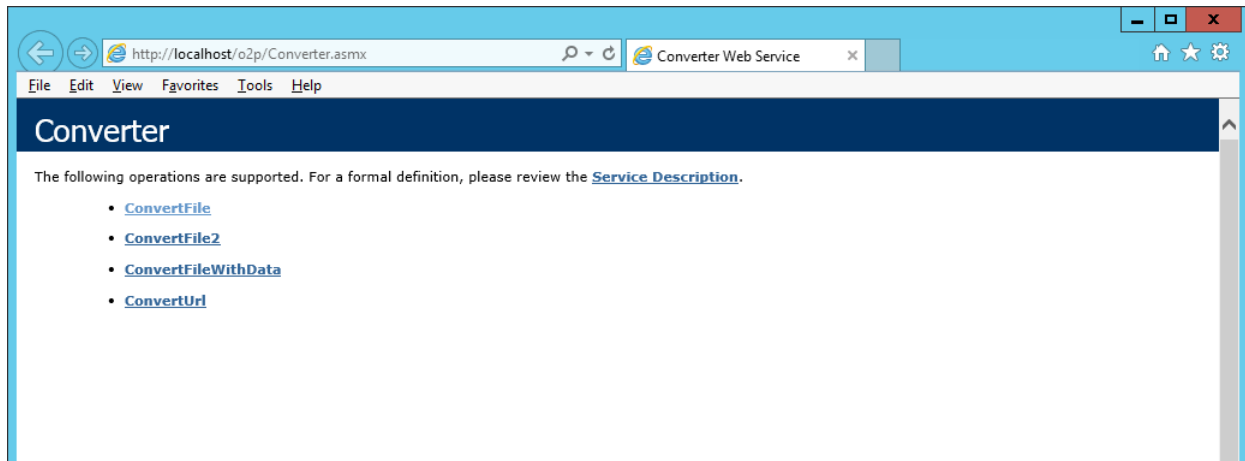
You can choose an arbitrary alias name (here o2p). The application pool you select must be configured for the Microsoft.NET 4.0 framework. On a Windows Server 2008, the default “DefaultAppPool” satisfies this requirement. (Consult [.NET Framework Versions and Dependencies](#) for more information.)



The web service is now ready for use. You can verify this by browsing Converter .asmx.



This should open a browser window as the one shown below.



**Note:** Being ASP.NET based, many parameters that are relevant to the web service are controlled via the `web.config` file. You may want to inspect this configuration file and make adjustments as required. See the appendix [Web.config](#) for a detailed sample.

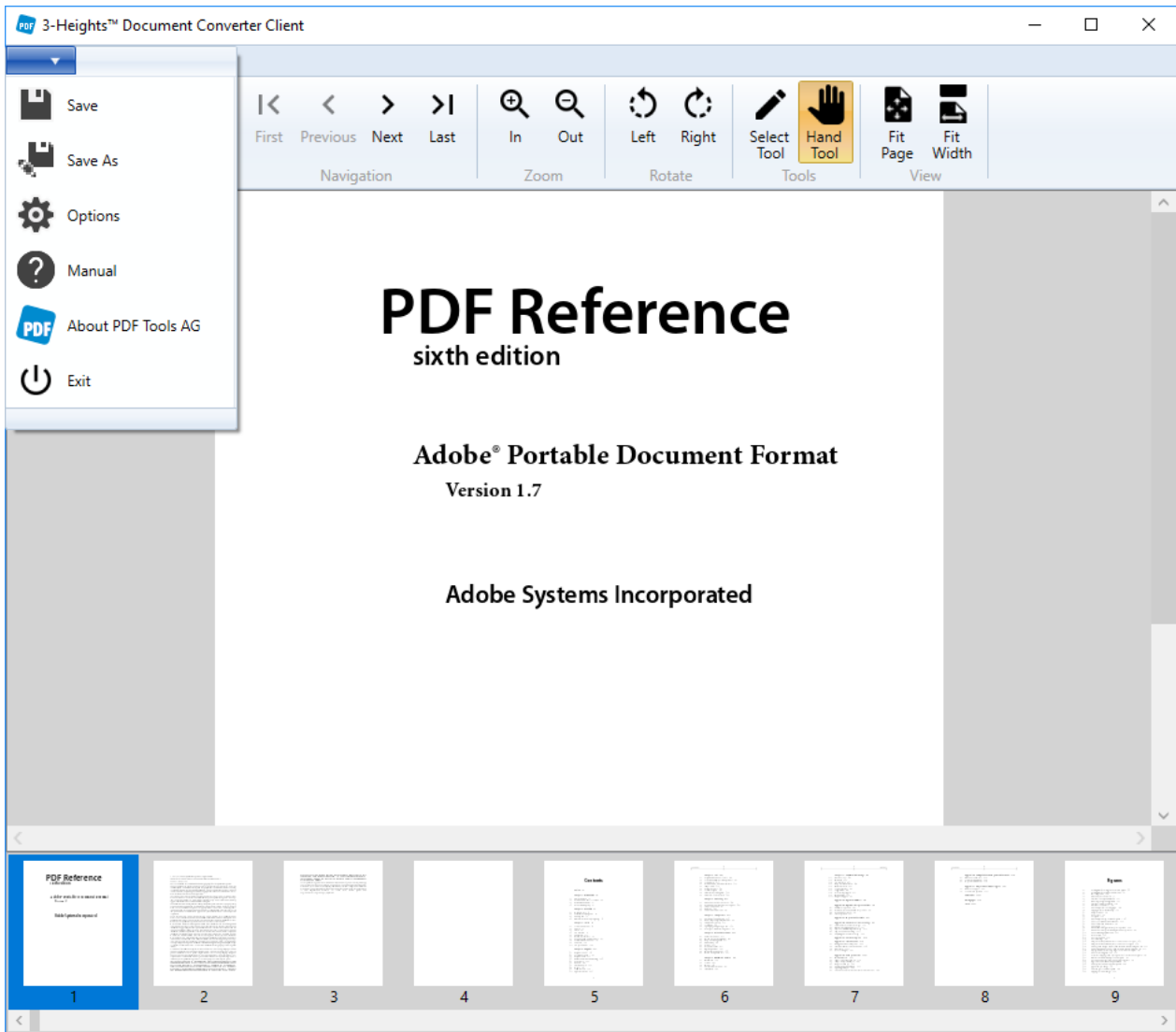
## 4.6.1 Using MTOM with the web service

The web service provided with the Document Converter installation can be enabled to support MTOM ("Message Transmission Optimization Mechanism"). The relevant information can be found on the [Microsoft support and download pages](#) related to "Web Services Enhancement (WSE) 3.0 for Microsoft.NET".

Essentially, enabling of MTOM consist of installing `Microsoft.Web.Services3.dll` and entering the appropriate entries into the `Web.config` file.

## 4.7 Client application

The 3-Heights® Document Converter Client offers a Graphical User Interface application which will send input document(s) to the 3-Heights® Document Converter service, either through the configured network port or using the Webservice interface. The output document can then be stored and/or displayed in the Client Application.



## 4.7.1 Features

**Viewer features** The built-in PDF viewer provides these features:

**Navigation** Using the arrow key, the navigation buttons or by selecting a page in the Thumbnails preview, a user can easily navigate through a document.

**Zoom** The viewer offers zoom in and out as well as **FitPage** and **FitWidth** modes, where the page size is automatically adjusted.

**Selection** Text can be selected and copied to the clipboard.

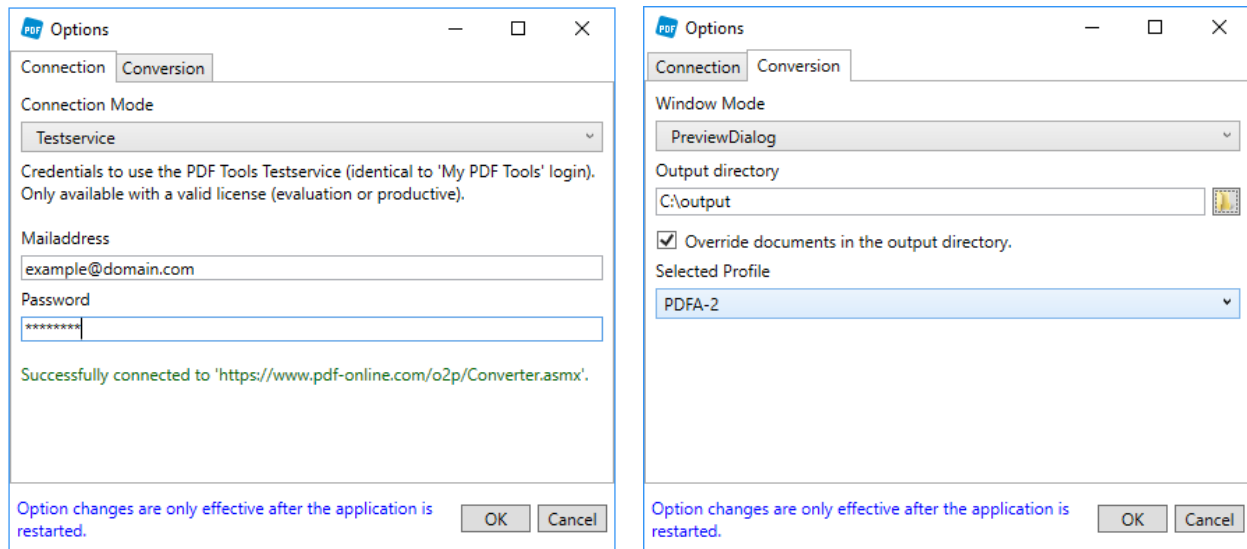
**Manipulation features** The Thumbnail Page Preview offers these manipulation features:

**Page order** One or several pages can be rearranged using Drag&Drop.

**Rotation** One or several pages can be selected and rotated either clockwise or counter-clockwise.

**Delete page** One or several pages can be removed from the document.

## 4.7.2 Options



The **Options** window allows to configure the connection to the 3-Heights® Document Converter service (**Connection** tab) and set the conversion options (**Conversion** tab). In the **Connection** tab, the connection status is displayed. Documents can only be converted if the client can successfully connect to a Document Converter service.

Connection options as well as the window mode are persisted on a per-installation basis, and are stored in a configuration file that is located in the installation folder. This configuration file also contains defaults for output directory, output override and selected profile that can be changed by users. These settings are persisted in the user profile.

### Connection Mode

**Key:** Connection Mode

**Network** Connect to your local server where the Document Converter Service is running using the network **hostname** of the server and the configured **port**. Ensure that no firewall blocks TCP traffic between the client and the server.

**Webservice** Connect to your server where the Document Converter Webservice is running.

**Testservice** Connect to the Pdftools Testservice. The credentials to use the Pdftools Testservice are identical to the “My PDF Tools” account information which you received after the registration process on [www.pdf-tools.com](http://www.pdf-tools.com). To access the Testservice, a valid license (evaluation or productive) needs to be available in your account.

### Window Mode

**Key:** Window Mode

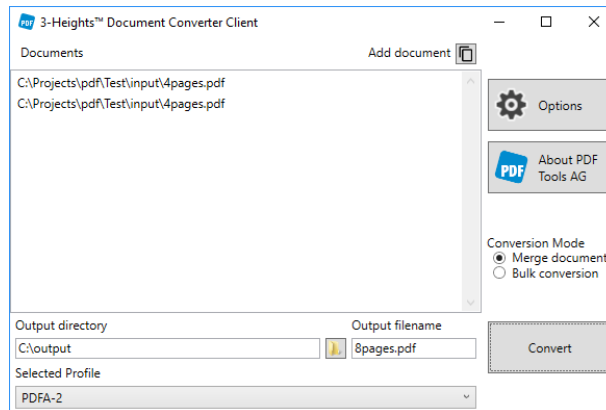
To change the Window Mode or other options, open the Client from the Windows Start Menu without initializing any conversion.

**NoDialog** Documents will be saved in the configured [Output directory](#) without any dialog window. If a file with the same name already exists in the directory, the current time is added to the filename.

**OutputDialog** Only a Windows Save Dialog window is shown so that the user can decide where to save to output document.

**PreviewDialog** A document preview is shown after the conversion so that the output document can be reviewed and manipulations are possible.

**CollectDialog** Documents can be assembled and then be sent to the conversion service. This allows to easily merge documents or bulk process files.



## Output directory

**Key:** Output directory

Select the default output directory for converted documents.

## Override

**Key:** Override

If this option is selected, existing documents in the output directory will be overridden. Otherwise, the output document name will be adjusted (e.g. appended with the current time) to ensure unique file names.

## Selected profile

**Key:** Selected profile

The profile for the conversion can be selected. Changes are only effective after a restart. Profiles and their options sets (as described in [Section 6.4](#)) can be edited and added in the configuration file `DocConvDialog.xml`

Predefined Profiles:

**PDF/A-2** Convert documents to PDF/A-2 for archiving and long-term preservation.

**PDF/A-3** Convert documents to PDF/A-3 for archiving and long-term preservation (with support for embedded files).

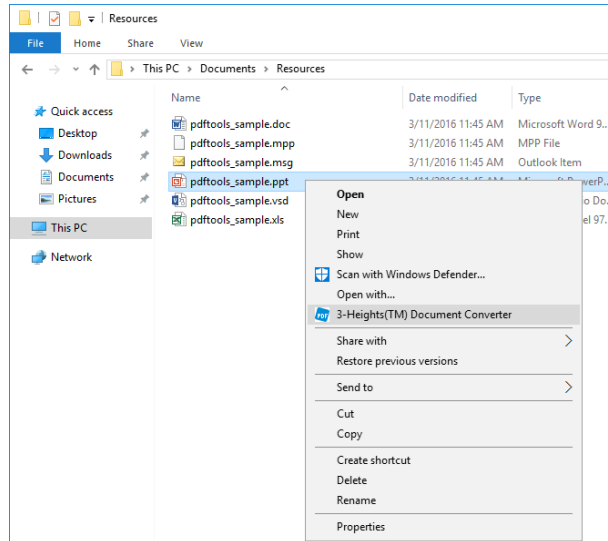
**PDF** Convert documents to PDF.

**OCR** Convert documents to PDF/A-2 and perform text recognition (OCR), recommended for scanned documents. Needs a configured OCR service to work.

### 4.7.3 Explorer shell extension

Another way to use the Shell Pop-up is by the Explorer Shell Extension.

Right-click on a document, the pop-up menu will show the “3 Heights(TM) Document Converter” entry.



The menu entry will be shown when one or multiple documents are selected, and the selection contains documents having a known file extension. It will also be shown for folders that contain corresponding files.

When activating the menu, it runs the Shell Pop-up.

## Configuration

The pop-up menu will be shown by the explorer for files with certain name extensions.

The list of relevant file extensions is configured via the `O2PCM.ini` (or `O2PCM64.ini`) file that is located in the installation directory of the Document Converter Client. See the table [Supported document extensions](#) for available extensions.

**Note:** The `EXPLORER.EXE` file needs to be restarted to reflect changes in the `O2PCM.ini` file.

### 4.7.4 Office add-in

The Office add-ins are a simple way for converting documents directly from an Office Application. They are available for these Office products:

- Microsoft Word
- Microsoft Excel
- Microsoft Outlook

## 4.8 Command line tool

The command line executable `o2pclient.exe` is a convenient tool for automating conversion tasks in a shell script or as a scheduled task.

Unless you have extended the `PATH` environment setting to include the installation directory of the Document Converter, you have to specify the full file path to call the `o2pclient.exe` program.

**Usage:** When called without any arguments, it will print a usage message to standard output

```
Usage: o2pclient [ Options ] office.doc office.xls ...
Options: -sp url      service point URL overriding entry in .config file;
          NOTE: this must be first option on the command line.
          -l          list known file extensions
          -o out.pdf  specify name of output file (default: out.pdf)
          -j options  set job options
          -b options  set file options
          -px n v.dat specify parameter name and data file
          -v          verbose
          -ax file    add XML Metadata from file (unchecked)
          -share      pass files via shared file system

Job options: PDF.USERPASS=pass;PDF.OWNERPASS=pass;PDF.PERMISSION=-1;
             PDF.DATE=20070901120000
File options: "Outline='Bookmark Text;';xz.zip:Outline=Archive XY"
```

The `o2pclient.exe` options `-b` and `-j` work analogue to the corresponding [Watched folder thread options](#). For a detailed list of available parameters see [Document options](#) and [Job options](#).

**Note:** The `o2pclient.exe` options `-j` and `-b` are merged with the options, which are set with the Service Configuration Editor. In case of conflicts `-j` and `-b` override the static ones.

Input documents may contain "\*" wildcard characters to easily pass multiple files for processing. The sequence will be determined by the lexical ordering as provided by the Windows file system.

**Example:** Convert all documents in a directory to a single PDF/A document

```
o2pclient -o folder.pdf -j PDF/A -b Outline=FILENAME .\folder\*.*
```

**Note:** You can specify a relative or an absolute path for input files on the command line. Please make sure that the extension of the output file corresponds to the actual document format you expect as output (e.g. out.tiff, out.zip)

# 5 Processing guide

## 5.1 PDF/A

The Document Converter supports creation and conversion of documents to produce PDF/A conforming output.

Essentially, the [PDF/A](#) job option must be set to produce PDF/A conforming output. If a job inputs include PDF/A documents, the Converter Service may need to override conflicting settings or adjust the conformance level to reflect the properties of the overall merged document.

PDF/A conversion at the job level is performed once the merge process is complete in a separate step by the Dispatcher service (O2PSRV.exe). Since PDF/A conversion requires the embedding of color profiles, these resources must be available to O2PSRV. The default color profiles used by the service are USWebCoatedSWOP.icc and sRGB Color Space Profile.icm. Make sure these files are located in the \spool\drivers\color sub-path below the Windows System directory (usually C:\WINDOWS\System32).

## 5.2 Stamping

PDF documents can be stamped while being processed. Stamp information is passed via an XML file. The XML file can be located on the server, or can entirely be passed via API. See in the appendix [Stamp file syntax](#) for a detailed description of the XML stamp file.

### 5.2.1 Pre-installed stamp files

Pre-installed stamp files are stored on the server, preferably in the installation folder of the converter service or a subfolder thereof. A pre-installed stamp file is applied via the `STAMP=stamp.xml` job option. `stamp.xml` is the name of the stamp file. This can be an absolute file path, or a path relative to the installation folder of the converter service.

It is possible to define place holders in the XML file that can be replaced with actual data.

**Example:** Specify place holders for `name` and `mail` in the `stamp.xml`

```
<?xml version="1.0" encoding="utf-8"?>
<pdfstamp xmlns="http://www.pdf-tools.com/pdfstamp/">
  <stamp page="all" align="center middle">
    <text font="Arial" size="25">[[name]] [[mail]]</text>
  </stamp>
</pdfstamp>
```

`STAMP=stamp.xml?name=John Smith&&mail=john.smith@moon.ext`

Any occurrences of `[[name]]` will be replaced with the actual string `"John Smith"` and `[[mail]]` will be replaced with the actual string `"john.smith@moon.ext"`.

**Note:** The converter service assumes the XML stamp file is encoded in UTF-8.

## 5.2.2 Passing entire stamp files via API

An entire stamp XML file can also be passed via the programming interfaces available (including the web service).

### Example:

```
res = ws.convertFileWithData (docBytes, // bytes of input document
    "STAMP=param:x;ORIGINALNAME=p.doc", // specify STAMP source ("x")
    xmpBytes, // XMP metadata (may be null)
    "x", // parameter name "x" (must match
        // stamp source)
    stampBytes); // stamp XML data (UTF8 encoded)
```

## 5.2.3 References to fonts and images

A stamp file will reference fonts for text stamps, and image files for image stamps. While fonts can be both be specified via their TrueType name and via a file name, images can only be specified via a file name.

Since there is no mechanism to pass image contents along with the stamp file, this means that image data must be stored locally on the server. Images can thus either be pre-installed, or passed from the application via a shared folder.

## 5.3 Document transformations

Document transformations offer a possibility to extend the document converter using third party tools or custom logic. The basic principle is to have the converter execute a custom script or executable whenever a document is about to be converted.

For example, a document transformation can be used to process an input document in a special way prior to further conversion via the standard built-in components of the converter service.

The converter service will perform transformations when the document option `TRANSFORM=value` is applied. The `value` string is either the name of a VBS script located in the installation directory of the converter service, or an arbitrary command that can be executed by Windows.

For security reasons, the commands that are permitted must be configured in the [Options](#) section of the `O2PWSC.ini` file:

```
[Options]
AllowedTransforms = xsl.vbs : doXml.exe : ApplyXSL.exe *
```

The value passed along with the `TRANSFORM` option will be validated against the value string for [AllowedTransforms](#). There must be an exact (case sensitive) match for the string part preceding the `*` character, otherwise, an error will be raised, and the conversion will fail.

The `*` placeholder allows you to pass additional parameters to the transformation tool which may vary from document to document. The colon character is used to separate valid alternatives (spaces before or after the separator are ignored).

The converter service will append the path name of the input document as the last command argument. When a VBS script is specified with the `TRANSFORM` option (e.g. `TRANSFORM=t1.vbs`), it is executed via `cscript.exe`. No extra parameters can be passed.

In any case, the transformation command shall write to output a line starting with "Output: ", followed by the full name path of the file that has been created as the result of the transformation. The converter service will continue by converting this file rather than the original input file. After conversion, the file will be deleted.

The transformation command has the option to not process the input document. In this case, it will write "Output: =input" to output.

If the transformation command fails to produce a conversion output, it can write a line starting with "Error: " followed by a descriptive error text.

It may also write a line starting with "Warning: " followed by a descriptive text. This line must appear before "Output: " to have any effect.

If you expect a transform command to take more than just a few seconds of time to execute, make sure to adjust the [MAXTRANSFORMTIME](#) setting.

## 5.4 Optimization

Optimization is an optional feature of the Document Converter, which can be applied using the [PDFOPTIMIZE](#) job option.

### To make use of this feature

**PDF Optimizer Shell** The product "3-Heights® PDF Optimizer Shell" needs to be installed and a separate license configured. After the installation process the Document Converter needs to be restarted.

**Choose a profile** Several predefined profiles such as [archive](#), [web](#) and [print](#) are available. Set a custom profile or change a predefined one with the [Service configuration editor](#) on the "Document Conversion Settings" tab via the PdfOptimize Scope.

**Set the path to pdfoptimize.exe** Set the executable path of the 3-Heights® PDF Optimizer Shell tool with the application option [exepath](#). After having installed this tool using the MSI installer kit, the System PATH environment variable will be extended.

**PDF Optimizer documentation** Carefully read the [PDF Optimizer Shell](#) documentation.

**Example:** Assume the [archive](#) profile is set; i.e. the O2PWSC.ini configuration file contains the following lines

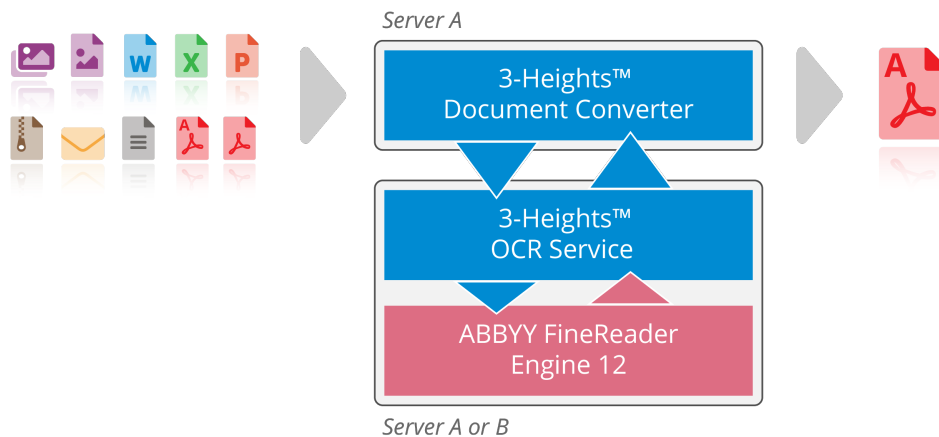
```
[PdfOptimize]
archive = -or -od -st -dt 200 -dr 200 -fb 6,7,10 -fc 1,8,10 -fi 2,10 -m -s -oc
exepath = pdfoptimize.exe
```

This allows you to specify [PDFOPTIMIZE=archive](#) in the job options string, having the effect that the command `pdfoptimize.exe -or -od -st .. <inpdf> <outpdf>` is executed, where `<inpdf>` is a placeholder for the intermediate PDF document obtained from the conversion process, prior to passing it to the last step involving PDF/A conversion and/or signing; `<outpdf>` is a placeholder for the file path where the optimized output shall be stored.

## 5.5 OCR

As an optional feature optical character recognition (OCR) can be activated in your 3-Heights® Document Converter configuration. Scans or images in digitally produced documents are made readable, and missing Unicode characters in embedded fonts are added so that this text is also readable. All recognized text is embedded in the document,

making it searchable. The text can also be extracted at any time using additional tools. To integrate an OCR engine in your Document Converter following additional products are necessary:



1. **ABBYY FineReader OCR engine** must be installed and licensed separately (OCR-FRxx-Installer.zip). Supported are:
  - 3-Heights® OCR Add-on for ABBYY FineReader Engine v11: [OcrAbbyy11.pdf](#)
  - 3-Heights® OCR Add-on for ABBYY FineReader Engine v12: [OcrAbbyy12.pdf](#)
2. **3-Heights® OCR Service:** [3-Heights® OCR Service](#) (Ocr-Service-x.x.x.x-Windows-(xxbit).msi)  
The Ocr-Service MSI installs a service that coordinates and parallelizes OCR jobs.  
→ It's strongly recommended to use the 64-Bit version of the OCR Service.

## 5.5.1 Use cases

### First steps

1. Specify [OCR.ENGINE](#)
2. Set [OCR.LANGUAGE](#)

### How to make text extractable

#### detect text contained in images:

For documents that contain images, processing of images can be activated by setting: [OCR.IMAGEMODE](#)

```
OCR.ENGINE=Service;OCR.LANGUAGE=English;OCR.IMAGEMODE=update
```

#### make text extractable:

For documents that contain non-extractable text, processing of text can be activated by setting: [OCR.TEXTMODE](#)

```
OCR.ENGINE=Service;OCR.LANGUAGE=English;OCR.TEXTMODE=update
```

#### make other visible text extractable:

For documents that contain other forms of visible text, pages can be OCR processed by setting: [OCR.PAGEMODE](#)

```
OCR.ENGINE=Service;OCR.LANGUAGE=English;OCR.PAGEMODE=all
```

## How to detect and embed barcodes into metadata

### Example:

1. Set [OCR.PARAMETERS](#) to a PredefinedProfile described in the **3-Heights® OCR Add-on for ABBY FineReader Engine manual**:
  - BarcodeRecognition\_Accuracy - for barcode extraction, optimized for accuracy
  - BarcodeRecognition\_Speed - for barcode extraction, optimized for speed
2. Set [OCR.PAGEMODE](#) to all - process all pages that are not empty
3. Set [OCR.EMBEDBARCODES](#) to true - embed barcode information into the document XMP metadata

```
OCR.ENGINE=Service;OCR.LANGUAGE=English;  
OCR.PARAMETER=PredefinedProfile=BarcodeRecognition_Accuracy;OCR.PAGEMODE=all;  
OCR.EMBEDBARCODE=true
```

## 5.6 Plugins

The Document Converter can be extended by custom plugins. These are DLLs that are configured to be loaded at startup. They must implement a set of functions to support the conversion of documents of the plugin specific format.

The SDK folder distributed with the Document Converter contains the C header file (`o2ppplugin.h`). Please refer to the documentation within the header file.

To activate any plugin, add the configuration line `Plugins = <name>.DLL` into the [Options](#) section of the `O2PWSC.ini` file. If there are multiple plugin DLLs, list them all on one line, separated by a space character. A plugin DLL shall be located in the installation directory of the Document Converter.

Plugin DLLs are loaded during initialization of the O2PWSC executable.

### 5.6.1 ScriptPlugin DLL

PDF-Tools AG provides a standard plugin DLL `ScriptPlugin.DLL` is a generic plugin that builds a bridge between the C interface and Visual Basic Script.

A sample script in VBS is also provided and illustrates how AutoCAD 2008 could be integrated. To activate this plugin, add the configuration line `Plugins = ScriptPlugin.DLL` into the [Options](#) section of the `O2PWSC.ini` file, and adjust the VB-script in `ScriptPlugin.script` according to your needs.

It is possible to load and use multiple plugin DLLs. If you have three different applications needing three different VB-scripts, just make copies of the `ScriptPlugin.DLL` and name it according to your preferences – e.g. `AutoCAD-Plugin.DLL`, and use the same names for the scripts (e.g. `AutoCAD-Plugin.script`). On the `Plugins=` line, just append the names of the additional plugin DLLs, separated by a white space, semicolon or colon.

### 5.6.2 NetPlugin DLL

A Microsoft.NET assembly can also host a custom plugin. It must implement the `IPlugin` interface and provide a constructor that takes an `IContainer` argument. These interfaces are defined in the `NetPlugin.DLL`, which must

be referenced. In addition, the name of the assembly must match the name pattern `*NetPlugin*.dll`, and `NetPlugin.DLL` must be listed in the `Plugins` line in the [Options](#) section as described above.

`NetPlugin.DLL` is installed by default into the installation directory of the Document Converter. A source code sample is provided on request.

# 6 Interface reference

## 6.1 Document type index

The Document Converter contains a number of standard modules for controlling external document authoring applications as well as built-in support for PDF and raster images.

Document types

Application/Format	INI-Section Name	Scope Name
Internet Mail Messages	<a href="#">Eml</a>	Eml
JPEG, TIFF, GIF, BMP, PNG images	<a href="#">Image</a>	Image
http/HTML based Web page	<a href="#">HTML</a>	HTML
Microsoft Excel	<a href="#">Excel</a>	Excel
Microsoft PowerPoint	<a href="#">PowerPoint</a>	PowerPoint
Microsoft Outlook	<a href="#">Outlook</a>	Outlook
Microsoft Word	<a href="#">MSWord</a>	MSWord
Microsoft Visio	<a href="#">Visio</a>	Visio
Microsoft Project	<a href="#">MSProject</a>	MSProject
OpenOffice	<a href="#">OO</a>	OO
PDF	<a href="#">PDF</a>	PDF
Text (ANSI, UTF-8, Unicode)	<a href="#">TXT2PDF</a>	TXT2PDF
Windows Enhanced Metafile	<a href="#">EMF</a>	EMF
XML Paper Specification	<a href="#">XPS</a>	XPS

## 6.2 Document extension index

The application used for converting a particular document is selected according to the document format. In a first attempt, document formats are determined based on the file extension. In most cases, the file extension corresponds to the actual document format. If the extension is unknown or incorrect, the 3-Heights® Document Converter tries to guess the document format from the file header, and finally tries to open the document with each application.

The following table lists the file extensions registered for the supported applications. Any plugin registered with the Document Converter extends this table with their proprietary extensions.

## Supported document extensions

Application	Extension
Enhanced Metafile	.emf .emz
http/HTML	.url .mht .htmzip
Internet Mail Messages	.eml
Microsoft Excel	.xls .xlt .xlsx .xlsm .xlsb .xltx .xltm
Microsoft Project	.mpp
Microsoft PowerPoint	.ppt .pps .pot .pptx .pptm .ppsx .ppsm .potx
Microsoft Outlook	.msg
Microsoft Word	.doc .docm .docx .dot .dotx .dotm .rtf .txt .htm .html .wpd .wpc .ws
Microsoft Visio	.vsd .vsdx .vsdm .vdx .vssx .vssm .vss .vsx .fstx .vstm .vst .vtx .vsw .vdw .svg .svgz
OpenOffice.org	.odf .odg .odp .ods .odt .sxw .sxi .sxc
Raster image formats	.jpg .jpeg .bmp .gif .tif .tiff .jb2 .jp2 .png .heic .heif
RAR Archives	.rar
Text	.txt .log .ini
XML Paper Specification	.xps
ZIP Archives	.zip

**Note:** The ZIP provider "7z" can be configured to support also archives having file extensions .7z, .bzip2, .gz, .tar, .wim and .zipx. Therefore set the key [PROVIDER=7z](#).

## 6.3 Application options

The Document Converter contains control modules for each document format that automates the corresponding office applications. There is a standard way of configuring these modules, which works with a windows profile file (application **INI** file).

The name of a configuration **INI** file corresponds to the file name of the executable. In this sense O2PWSC.exe will read from O2PWSC.ini.

**Note:** O2PSRV.exe does not directly control any office applications; therefore, there is no need for an O2PSRV.ini file.

See chapter [User guide](#) for configuration guidance of the service and the different Document Converter features.

**Example:** An INI-file (O2PWSC.ini)<sup>2</sup>

```
[Outlook]
OUTLOOKEXE=\Microsoft Office\Office12\OUTLOOK.EXE
button_yes=&Ja
access_grant = &Zugriff
[MSWord]
preload = false
SAVEASPDF = true
WorkingSet = 90M
```

Each configuration setting applies to the section for the specific office application (Microsoft Word and Outlook in the above sample).

**Note:** INI-Section and setting names are not case sensitive, and generally, values are neither, unless specified.

### 6.3.1 Common to all external applications

The INI Keys described in this subsection are synonymously applicable for all external applications.

#### disabled

**Key:** `disabled` Default: `false`

**true** Disable (do not use) the application.

**false** Do not disable the application.

#### Extensions

**Key:** `Extensions` Default: `(application specific)`

Replace the standard file extensions for the application. This is a static option (i.e. it cannot be changed or overridden at runtime).

Syntax: list of colon separated extension strings.

#### Example:

```
[MSWord]
Extensions=DOC:DOCX:RTF
```

<sup>2</sup> Certain values for MS Outlook are language specific, see [OUTLOOKEXE](#), [button\\_yes](#), [access\\_grant](#)

## MaxCallDuration

**Key:** `MaxCallDuration` Default: `600`

Time limit in seconds allowed for calls into an office application. If this time limit is exceeded, the application is forcibly terminated.

The MaxCallDuration option also applies to a number of processing task, such as PDF optimization, XFA rendering via Acrobat Reader or OCR processing. Defaults are task specific and can be configured in O2PWSC.ini or passed via document or job options (e.g. pdfocr:MaxCallDuration=3600).

## preload

**Key:** `preload` Default: `false`

**true** Start the application during initialization to have it ready when needed to process a document.

**false** Do not start the application during initialization.

## RestartAfterConversions

**Key:** `RestartAfterConversions` Default: `(unlimited)`

Number of conversions till the office application is restarted by the service.

Use this configuration option to work around instabilities.

## RestrictToExtension

**Key:** `RestrictToExtension` Default: `false`

**true** Only attempt to open documents having one of the application specific file extensions.

**false** Do not restrict to open documents having one of the application specific file extensions.

## WorkingSet

**Key:** `WorkingSet` Default: `100M`

Working set limit triggering automatic recycling of application process. Specify an integer number, optionally followed by **k** (for kilo bytes) or **M** (for megabytes).

## 6.3.2 [MSWord] INI-File Section

### BFFValidate

**Key:** `BFFValidate` Default: `strict`

Input document can be validated using the Microsoft BFFValidator. Corrupt documents can lead to Office application crashes, which will delay other conversions, or produce output documents with missing parts.

**strict** Fail if a Word-97-2003-Document is not a valid.

**warn** Warn if a Word-97-2003-Document is not valid, try to convert the document anyway.

**disable** Disable the validation, there will be no validation warning or error.

### BitmapMissingFonts

**Key:** `BitmapMissingFonts` Default: `true`

When using [SAVEASPDF](#), embed bitmaps for missing fonts.

### CreateBookmarks

**Key:** `CreateBookmarks` Default: `1`

When using [SAVEASPDF](#), use a specified value for the CreateBookmarks option.

- 0** none
- 1** headings
- 2** all Word bookmarks

### DocStructureTags

**Key:** `DocStructureTags` Default: `true`

When using [SAVEASPDF](#), embed document structure tags (required for PDF/A1a and PDF/A/UA)

### EmbeddedDocuments

**Key:** `EmbeddedDocuments` Default: `false`

**true** Convert also all embedded documents.

**false** Do not convert embedded documents.

## LockFields

**Key:** LockFields Default: false

**true** MS Word locks all fields in the document as well as headers and footers. This will prevent MS Word from updating their values at print time or on saving to PDF.

**Note:** MS Word updates certain fields before they can be locked, such as TIME (current date and time).

**false** Do not lock fields in the document.

## UpdateFieldsAtPrint

**Key:** UpdateFieldsAtPrint Default: false

**true** set the corresponding Word option to update fields before printing. Use this option in combination with SAVEASPDF=false (i.e. use Word's PrintOut function to convert the document).

**false** Do not set the corresponding Word option to update fields before printing.

**Note:** See also [LockFields](#)

## PW

**Key:** PW Default: ""

The password to open password protected documents. This option is usually specified per document.

## ShowComments

**Key:** ShowComments Default: false

**true** Make comments visible in converted document. This setting is only effective, if [SAVEASPDF](#) is false.

**false** Hide comments in converted document.

**Note:** Explicitly showing or hiding comments requires that the document be opened with `ReadOnly=false`. This may require a password. It is therefore recommended to not use this option unless explicitly required (and the password is known or known to not be required).

## SAVEASPDF

**Key:** `SAVEASPDF` Default: `true`

When option is not defined, `false` will be used.

**true** Use the Office “ExportAsFixedFormat” feature to produce a PDF from the Word document. Setting this value to `true` requires MS Office installed. It is required to create Bookmarks for the document titles.

**false** Use the application’s printing function for conversion via the 3-Heights® Document Converter Service PDF Printer.

This is also a document processing option (see [Job options](#)).

## UseISO19005\_1

**Key:** `UseISO19005_1` Default: `false`

When option is not defined, `true` will be used.

**true** When using [SAVEASPDF](#), produce PDF/A-1 conforming output.

## VBAOFF

**Key:** `VBAOFF` Default: `true`

When option is not defined, `false` will be used.

**true** Turns VBA off for MS Office applications by setting the VBAOFF entry in the registry of the worker account user.

**false** Turns VBA on for MS Office applications by setting the VBAOFF entry in the registry of the worker account user.

See [VBAOff setting has no effect](#) in the chapter [Troubleshooting](#).

## Overriding settings

The following option settings override settings made for a specific document and are intended for use on the job or document level. They are useful for non-native document formats like text or HTML.

## PageHeight

**Key:** PageHeight Default: 842 pt/A4

Page height in points (overrides [PaperSize](#)).

## PaperMargins

**Key:** PaperMargins Default: 72

Set all four page margins to the specified value. Intended for use at the document level.

## PageOrientation

**Key:** PageOrientation

Possible values are:

- 0 Portrait
- 1 Landscape

## PaperSize

**Key:** PaperSize

Set the page size. Possible values are the MSWord.WdPaperSize integers.

See [WdPaperSize Enumeration](#)

## PageWidth

**Key:** PageWidth

Page width in points (overrides [PaperSize](#)).

## PRINTMARKUPS

**Key:** PRINTMARKUPS

Controls the items of the document to be printed. Possible values are:

- A automatic (document default)
- D document content

## SoftErrFail

**Key:** `SoftErrFail` Default: `false`

Controls the handling of document related issues (soft errors) such as detection document corruptions during opening a document. The default behavior when this setting is undefined or set to false is to ignore soft errors and proceed with the conversion.

Possible values are:

**all** fail on any soft error

**foo|bar|corrupt** fail on soft errors having a description string containing one of the items of the vertical bar separated list of keywords

## LoadAllPages

**Key:** `LoadAllPages` Default: `false`

Force loading of all pages of the document to ensure the loading of large images prior to exporting or printing.

Enabling this option is generally not necessary and will significantly slow down the conversion speed, but advised when placeholder images appear in the output instead of the correct image representation.

### 6.3.3 [Excel] INI-File Section

#### SAVEASPDF

**Key:** `SAVEASPDF` Default: `false`

**true** Use the Office "ExportAsFixedFormat" feature to produce a PDF from the Excel document.

**false** Use the application's printing function for conversion via the 3-Heights® Document Converter Service PDF Printer.

#### BFFValidate

**Key:** `BFFValidate` Default: `strict`

Input document can be validated using the Microsoft BFFValidator. Corrupt documents can lead to Office application crashes, which will delay other conversions, or produce output documents with missing parts.

**strict** Fail if a Excel-97-2003-Documents is not a valid.

**warn** Warn if a Excel-97-2003-Documents is not valid, try to convert the document anyway.

**disable** Disable the validation, there will be no validation warning or error.

## FailPageCount

**Key:** `FailPageCount` Default: `4001`

Raise the "O2P\_W\_PARTSMISSING" warning for a sheet that prints the specified number of pages or more (0 unlimited). The behavior of this option changed as of version 4.9.14.0; previously, the document would fail with a "O2P\_W\_SOURCEQUALITY" warning, and no pages of the worksheet in the output. The new behavior is that up to FailPageCount pages are converted, with an unknown number of further pages missing.

## FitToPage

**Key:** `FitToPage` Default: `false`

**true** Try built-in heuristics to use either Excel's "FitToPagesTall" or "FitToPagesWide" PageSetup property depending on the information available from PrintArea or UsedRange.

**Note:** This setting is only applicable if the resulting downscaling is not too high (limited to 50 %).

**numeric value** Specify a numeric value between 10 and 99 to enable page fitting with a maximum downscale percentage. The numbers may be followed by the letter `l` to apply "width" fitting.

**false** Do not modify the document's page setup parameters, i.e. convert with the settings stored with the document.

Setting for Print Options (see [Job options](#)).

## ForceLetter

**Key:** `ForceLetter` Default: `true`

**true** Changes Letter format to Letter-small during conversion to preserve Letter format in case the standard paper size on the printer differs.

**false** Excel will use the format of the standard printer instead of Letter (undocumented "feature").

## MaxPages

**Key:** `MaxPages` Default: `"all pages"`

Print up to the specified number of pages for a sheet.

## PaperSize

**Key:** PaperSize

Override the paper size stored with the workbook.

Specify a numeric value according to the paper size definitions for Windows, e.g. 1 for letter, 9 for A4. See [Microsoft Documentation](#)

It is possible to use this setting as document option, and to prefix it with the sheet number, e.g. "Sheet1:PaperSize=8"

## Orientation

**Key:** Orientation

Override the page orientation stored with the workbook. Possible values are P (portrait) or L (landscape).

This setting can also be used as a document option, optionally prefixed with the sheet number (e.g. Sheet1:Orientation=L)

## PrintArea

**Key:** PrintArea

By default, use the PrintArea stored with the sheet. If none is specified, UsedRange determines the area to be converted.

You can pass a value to the PrintArea option to specify a PrintArea to be used in case of the PrintArea of the sheet being empty, or to replace it.

**Examples:** Use A1:H4 if sheet has no PrintArea defined.

```
PrintArea=A1:H4
```

Use A1:H4, replacing any PrintArea that may be defined with the sheet.

```
PrintArea==A1:H4
```

Use default behavior of Excel for printing, i.e. PrintArea if defined, Excel's heuristics otherwise.

```
PrintArea=
```

## PW

**Key:** PW Default: ""

The password to open password protected documents. This option is usually specified per document.

## SHEET

**Key:** SHEET Default: all

Sheet(s) to convert. The values that can be specified are

**active** The active Excel sheet.

**all** All Excel sheets.

**<page number>** The number of a specific Excel sheet (e.g. 1 for the first sheet).

**Note:** This setting can also be passed as a document option.

## TIFF.DPI

**Key:** TIFF.DPI

Set the PrintQuality in DPI (dots per inch) to be applied in the Page Setup for printing. If left empty, the setting stored with the sheet or chart is used (default).

### 6.3.4 [PowerPoint] INI-File Section

#### BFFValidate

**Key:** BFFValidate Default: strict

Input document can be validated using the Microsoft BFFValidator. Corrupt documents can lead to Office application crashes, which will delay other conversions, or produce output documents with missing parts.

**strict** Fail if a PowerPoint-97-2003-Document is not a valid.

**warn** Warn if a PowerPoint-97-2003-Document is not valid, try to convert the document anyway.

**disable** Disable the validation, there will be no validation warning or error.

#### BitmapMissingFonts

**Key:** BitmapMissingFonts Default: true

When using [SAVEASPDF](#), embed bitmaps for missing fonts.

## DocStructureTags

**Key:** `DocStructureTags` Default: `true`

When using [SAVEASPDF](#), embed document structure tags (required for PDF/A1a and PDF/A/UA)

## FitToPage

**Key:** `FitToPage` Default: `false`

Setting for Print Options. (This is also a document processing option; see [Job options](#))

## OutputType

**Key:** `OutputType` Default: `1`

Print Options setting

- 1 Slides
- 2 TwoSlideHandouts
- 3 ThreeSlideHandouts
- 4 SixSlideHandouts
- 5 NotesPage
- 6 Outline
- 7 BuildSlides
- 8 FourSlideHandouts
- 9 NineSlideHandouts

(This is also a document processing option; see [Job options](#))

## PrintComments

**Key:** `PrintComments` Default: `false`

Print comments and markups when converting using the print function (i.e. this option has no effect if using [SAVEASPDF=true](#)).

## SAVEASPDF

**Key:** `SAVEASPDF` Default: `true`

When option is not defined, **false** will be used.

**true** Use the Office "ExportAsFixedFormat" feature to produce a PDF from the PowerPoint document.

**false** Use the application's printing function for conversion via the 3-Heights® Document Converter Service PDF Printer.

### UseISO19005\_1

**Key:** UseISO19005\_1 Default: **false**

**true** When using [SAVEASPDF](#), produce PDF/A-1 conforming output.

## 6.3.5 [Visio] INI-File Section

### SAVEASPDF

**Key:** SAVEASPDF Default: **true**

When option is not defined, **false** will be used.

**true** Use the Visio "ExportAsFixedFormat" feature.

**false** Use the application's printing function for conversion via the 3-Heights® Document Converter Service PDF Printer.

### SetActivePrinter

**Key:** SetActivePrinter Default: **false**

Controls use of the "ActivePrinter" application property of Visio. By default, this property is not used, but the system's default printer is set.

**Note:** Some Versions of Visio may run unstable when using this feature.

### USEEMF

**Key:** USEEMF Default: **false**

**true** Use Visio's "SaveAs EMF" feature.

## VisioPrintAll

**Key:** `VisioPrintAll` Default: `false`

Controls whether “print all pages” is used.

**false** Has the effect that the number of non-template pages is determined, and the pages to be printed is specified continuous when using Visio’s Print command (work-around for a Visio 2003 bug).

## 6.3.6 [Outlook] INI-File Section

**Note:** Outlook is disabled by default in this version of Document Converter (`disabled=true`). The `msg2eml` component for Outlook mails (.msg) will be used instead. All options in the `Outlook` INI-file section will be ignored and the options set in the `Em1` INI-file section will take effect.

If you enable Outlook (`disabled=false`), please ensure that Outlook is configured for all Worker Users as described in section [Outlook configuration details](#) and the options are configured in this `Outlook` INI-file section.

## access\_grant

**Key:** `access_grant` Default: `&Zugriff`

The caption of the text string shown in the Outlook pop-up message (see above).

## button\_yes

**Key:** `button_yes` Default: `&Ja`

The caption of the Outlook pop-up message asking for grant to access email address information, this popup needs to be identified and automatically closed by the Document Converter.

## OUTLOOKEXE

**Key:** `OUTLOOKEXE` Default: `OUTLOOK.EXE`

Executable path of the Outlook application; this is needed to identify the process being controlled.

## SELECTFILES

**Key:** `SELECTFILES` Default: `.pdf .doc .xls .jpg`

See [Em1](#) INI-file section.

## SKIPFILES

**Key:** SKIPFILES Default: .db .xml

See [Em1](#) INI-file section.

## UseHtmlBody

**Key:** UseHtmlBody Default: false

**true** Save the body in HTML or RTF format to convert, rather than using the “PrintOut” function of Outlook.

**Advantage:** The page format will be adjusted as necessary for very wide pages in HTML format.

## 6.3.7 [MSProject] INI-File Section

### Chart

**Key:** Chart Default: Gantt Chart

Name or comma separated list of names of the view(s) to be converted.

**Note:** MS Project does not print views when they contain no data.

### DocStructureTags

**Key:** DocStructureTags Default: true

When using [SAVEASPDF](#), embed document structure tags (required for PDF/A1a and PDF/A/UA)

### PercentScale

**Key:** PercentScale Default: 100

Scaling factor in percent (no effect when [SAVEASPDF=true](#))

## PjPaperSize

**Key:** PjPaperSize Default: 8 (A3)

Paper size. Corresponds to Windows paper sizes.

See [Microsoft Documentation](#)

## Portrait

**Key:** Portrait Default: false

Paper orientation

## UseISO19005\_1

**Key:** UseISO19005\_1 Default: true

**true** When using [SAVEASPDF](#), produce PDF/A-1 conforming output.

## SAVEASPDF

**Key:** SAVEASPDF Default: true

When option is not defined, **false** will be used.

**true** Use the MS Project "ExportAsFixedFormat" feature.

**false** Use the application's printing function for conversion via the 3-Heights® Document Converter Service PDF Printer.

## 6.3.8 [00] INI-File Section

OpenOffice.org files

## SAVEASPDF

**Key:** SAVEASPDF Default: false

**true** Use the OpenOffice "PDF\_export" feature rather than printing via the 3-Heights® PDF Producer.

**false** Use the application's printing function for conversion via the 3-Heights® Document Converter Service PDF Printer.

## 6.3.9 [Em1] INI-File Section

Email messages (.eml files)

### HeaderTemplate

**Key:** HeaderTemplate Default: (built-in)

Specify the template to be used for rendering email headers.

The value either shall contain the template string in HTML format, or the name of a file containing that template prefixed with an "@" sign (default directory is the installation directory; a sample template file is provided)

This option can also be set as a document or job option, e.g. to select a language specific template. The value specified can be a file name (relative to the installation folder) or the HTML code itself (recognized by the presence of a <table> element).

The built-in header template for English is used by default. Two extended header templates for English and German are available in the installation folder. You may use these templates as-is, or customize them according to your needs, saving the template under different name. The elements of the sample templates should be self-explanatory. Note that entire lines in the template will be omitted when the header field is not present (e.g. the line containing the [[BCC]] placeholder should never show for real-life emails).

### Styles

**Key:** Styles Default: (built-in)

Specify the name of a file containing the HTML CSS styles to be used for formatting email content. A sample template file is provided in the installation directory.

### SELECTFILES

**Key:** SELECTFILES Default: .pdf .doc .xls .jpg

Space separated list of file extensions. Only attachments having their extension listed in this value will be converted.

### SKIPFILES

**Key:** SKIPFILES Default: .db .xml

Space separated list of file extensions. Attachments having an extension listed in this value will be ignored.

#### Example:

```
SKIPFILES=.db .exe .dll
```

**failing** Can be used to suppress attachments that cannot be converted successfully.

**none** Can be used to refer to file names without any extension in their name.

**SIZE<<width>x<height>** Small images can be filtered out by setting a value like **SIZE<4x5>**, i.e. the width must be at least 4 pixels, and the height at least 5 for the image to be converted.

### SkipUnusedInline

**Key:** `SkipUnusedInline` Default: `false`

**true** Images that are not embedded in the body will not be converted like attachments. Some mail clients show such images as attachment, others do not.

**false** Images that are not embedded in the body will be converted like attachments.

### DateFormat

**Key:** `DateFormat` Default: `dd.MM.yyyy HH:mm:ss' (z)'`

Format string to be used for date conversion.

The date format can also be set as a document or job option (with scope Eml).

A culture code placed in brackets can be optionally prefixed to the date format, e.g. "[en] MMM dd, yyyy hh:mm tt"

## 6.3.10 [TXT2PDF] INI-File Section

**Note:** If **TXT2PDF** is enabled, the extensions `.txt`, `.log` and `.ini` are no longer associated with MS Word (unless done so explicitly in the [MSWord](#) INI-file section with the key [Extensions](#) in the `O2PWSC.ini` file).

### TEXT.BORDER

**Key:** `TEXT.BORDER` Default: `20`

Border size in points (72 point = 1 inch).

## TEXT.FONTNAME

**Key:** TEXT.FONTNAME Default: CourierNew

Font name.

## TEXT.FONTSIZE

**Key:** TEXT.FONTSIZE Default: 11

Font size in points.

## TEXT.HEIGHT

**Key:** TEXT.HEIGHT Default: 842

Page height (pt/A4).

## TEXT.WIDTH

**Key:** TEXT.WIDTH Default: 595

Page width (pt/A4).

## TEXT.WRAP

**Key:** TEXT.WRAP Default: true

Wrap long lines.

## TEXT.LANG

**Key:** TEXT.LANG

Language code according to RFC 3066 (2 letter code according to ISO 639-1 if available, or three letter code according to ISO 639-2)

## TEXT.TITLE

**Key:** TEXT.TITLE

Document title string to set in PDF metadata

### 6.3.11 [EMF] INI-File Section

#### AutoRotate

**Key:** AutoRotate Default: true

Change paper orientation to landscape if EMF is landscape (and vice versa).

#### Border

**Key:** Border Default: 25

Value for the border margin in points (72 points = 1 inch)

#### PaperSize

**Key:** PaperSize Default: Printer setting

The Windows paper size value overriding the default paper size of printer. Please refer to the Microsoft documentation [Microsoft Documentation](#)

### 6.3.12 [PDF] INI-File Section

#### PDFA-PRECONVERT

**Key:** PDFA-PRECONVERT Default: false

**true** When merging an existing PDF document, convert it to PDF/A prior to merging.

**false** Do not convert to PDF/A prior to merging.

This is also a [Document options](#).

## URLPerms

**Key:** URLPerms

Adobe Reader setting for allowing or blocking access to web URLs.

- 0 Use custom settings.
- 1 Block all.
- 2 Allow all.

## RenderXFA

**Key:** RenderXFA Default: true

This option controls how PDF documents containing XFA forms are handled.

- true** Always use Adobe Reader to render XFA form PDFs. This setting is recommended to solve issues with PDFs coming from creators failing to generate correct field appearances in the PDF.
- false** Only use Adobe Reader to render XFA form PDFs if the NeedsRendering flag is specified, and the output shall be PDF/A.

### 6.3.13 [Image] INI-File Section

Settings applying to raster image input files (TIFF, JPEG, BMP, GIF, PNG, etc.) [disabled](#), [RestrictToExtension](#) or [Extensions](#).

The following configuration options can also be used as job or document options to control the conversion of individual documents.

## IMG2PDF . EXE

**Key:** IMG2PDF . EXE Default: false

**true** Perform image to PDF conversions via external invocation of the img2pdf.exe shell tool.

Use this option to reduce CPU and memory resources consumption within the service process and increase its stability.

Note: the converter aborts the image to PDF converter process when it takes longer than [MaxCallDuration](#) seconds (default: 600 or ten minutes). It is possible to change that period via job or document option by setting the option key `Image:MaxCallDuration` (or just `MaxCallDuration`).

**false** Perform image to PDF conversions in-process.

## FitImage

**Key:** FitImage Default: false

Values **A4**, **A4\***, **Letter**, **595x842\*** etc., **false**.

Set the page dimensions for PDF pages created from image documents. The meaning of the asterisk (\*) is: do automatically adjust the page orientation to the orientation of the image.

## Border

**Key:** **Border** Default: **0**

Define the width of a white border around the image in pages of the PDF document. The units are points (1point=1/72inch). The default is 0 points (no border, the image may fill the entire page).

## CMPRQUAL

**Key:** **CMPRQUAL** Default: **95**

Compression quality (**1**, ..., **100**) to be applied for lossy (re-)compression during Image to PDF conversion.

## ResolutionDPI

**Key:** **ResolutionDPI**

The image resolution in dots per inch (DPI) for downsampling images during conversion to PDF. The default is to not downsample images.

When [ThresholdDPI](#) is also specified, downsampling is performed only if the current resolution is above that threshold.

## ThresholdDPI

**Key:** **ThresholdDPI**

The DPI threshold for downsampling images during conversion. See [ResolutionDPI](#).

## FailSize

**Key:** **FailSize** Default: **32767**

Specify a size limit for images (maximum width or height) to avoid issues with extremely sized images.

## 6.3.14 [JPM] INI-File Section

### REPAIRSTREAMS

**Key:** REPAIRSTREAMS Default: true

When converting JPM documents to PDF, verify the integrity of the image streams, and repair as necessary.

## 6.3.15 [XPS] INI-File Section

### MaxCallDuration

**Key:** MaxCallDuration Default: 600

See [Common to all external applications](#); with XPS this option is also accepted as document processing option.

## 6.3.16 [PdfPrinter] INI-File Section

### Match

**Key:** Match Default: WORKER\_ID

Use only 3-Heights® PDF Producer printers which have the specified match string in their name.

Use this setting to ensure that the output from applications that printing to the default printer from multiple worker sessions cannot be safely identified.

An alternate method to ensure output separation is to use distinct user accounts and enforce single access via file security on the printer port directory.

### PaperSize

**Key:** PaperSize Default: location dependent (US, Canada: 1 (Letter); other: 9 (A4))

The Windows paper size value overriding the default paper size of printer. Please refer to the Microsoft documentation [Microsoft Documentation](#)

The location is taken from the windows system region's home location setting.

## Resolution

**Key:** Resolution

Set the resolution in dots per inch (DPI).

## Quality

**Key:** Quality Default: 80

Specify the JPEG compression quality between 1 - 100.

### 6.3.17 [TIFFPrinterBW] INI-File Section

## PaperSize

**Key:** PaperSize

The Windows paper size value overriding the default paper size of printer. Please refer to the Microsoft documentation [Microsoft Documentation](#)

## Compression

**Key:** Compression Default: 4 (CCITT G4)

Set the image compression. Possible values are 1 (None), 32946 (flate), 5 (LZW), 2 (CCITT G3), 3 (CCITT G3-2D), 4 (CCITT G4), 32773 (Packbits), and 34715 (JBIG2).

## ResolutionX

**Key:** ResolutionX Default: 200

Specify the document resolution in horizontal direction in DPI between 50 - 1200.

## ResolutionY

**Key:** ResolutionY Default: 200

Specify the document resolution in vertical direction in DPI between 50 - 1200.

## Quality

**Key:** `Quality` Default: `75`

Set the compression quality between `1` - `100`.

### 6.3.18 [TIFFPrinterColor] INI-File Section

## PaperSize

**Key:** `PaperSize`

See [PaperSize](#) in [TIFFPrinterBW].

## Compression

**Key:** `Compression` Default: `5 (LZW)`

Set the image compression. Possible values are `1 (None)`, `7 (JPEG Tech. Note #2)`, `32946 (ZIP)`, `5 (LZW)`, `6 (TIFFJPEG)`, `32773 (Packbits)` and `34712 (JPEG2000)`.

## ResolutionX

**Key:** `ResolutionX` Default: `200`

See [ResolutionX](#) in [TIFFPrinterBW].

## ResolutionY

**Key:** `ResolutionY` Default: `200`

See [ResolutionY](#) in [TIFFPrinterBW].

## Quality

**Key:** `Quality` Default: `75`

See [Quality](#) in [TIFFPrinterBW].

## 6.3.19 [HTML] INI-File Section

### IEPRINT

**Key:** IEPRINT Default: true

**true** Use the Internet Explorer browser control's print function to convert HTML documents. This is a static option.

**false** HTML documents will be converted via MS Word (unless file extensions are explicitly configured differently).

**Note:** When setting this to **true**, make sure to configure "Internet Explorer Enhanced Security" to "Off".

### HTMLMAXLOADTIME

**Key:** HTMLMAXLOADTIME Default: 30

Time limit for loading a HTML page (seconds). If this time limit is exceeded, IEPRINT is forcibly terminated and an error occurs.

### PageSetup

**Key:** PageSetup

Set parameters for Internet Explorer printing. The elements that can be set are the ones that are available via the File->Page setup menu in Internet Explorer. Background information: Internet Explorer stores the page setup on a per user basis in the registry. You may thus make any adjustments via IE dialogue, and then use the values as stored in the registry under HKCU/Software/Microsoft/Internet Explorer/PageSetup

**bottom=0.5** Set the bottom margin (units are inches).

**top=0.5** Set the top margin (units are inches).

**left=0.7** Set the left margin (units are inches).

**right=0.7** Set the right margin (units are inches).

**header=&w** Set the header (&w is a place holder for Title).

**footer=&u** Set the footer (&u is a place holder for URL).

**Print\_Background=yes** Print the page's background.

**Shrink\_To\_Fit=yes** Shrink the page to fit to the default paper size.

#### Example:

```
PageSetup = bottom=0.5,top=0.5,left=0.7,right=0.7,Background=Yes,header=,footer=
```

Note: item names can be abbreviated (e.g. bottom instead of margin\_bottom).

## HTMOPT

**Key:** HTMOPT Default: true

Optimize the HTML (or MHT) document to be converted.

This processing step includes the download of external images embedded in the HTML document, and enabling line breaks for long text lines. Very large images are scaled to fit the page dimensions.

**true** Enable HTML optimizations.

**false** Disable HTML optimizations.

**t:30** Set the timeout value applied to image downloads (seconds). Images that fail to download within this time frame will be replaced by a place holder. A value of 0 can be specified to disable downloading of images.

**w:712** Set the target width for scaling down oversized images.

**h:800** Set the target height for scaling down oversized images.

### Example:

```
HTMOPT=t:10,w:600,h:800
```

## 6.3.20 [ZIP] INI-File Section

### SKIPFILES

**Key:** SKIPFILES Default: .db .exe .dll .class .pif .lnk .lib .obj none

Space separated list of file extensions. ZIP and RAR archive members having an extension listed in this value will be ignored. E.g. .db .exe .dll

**failing** This special value can be used to suppress files that cannot be converted successfully.

**none** This special value can be used to refer to file names without any extension in their name.

**hidden** This value can be used to skip files having a period as the first character of the name.

### SELECTFILES

**Key:** SELECTFILES Default: .pdf .doc .xls .jpg

Space separated list of file extensions. Only archive members having their extension listed in this value will be selected for conversion.

## PROVIDER

**Key:** PROVIDER Default: 7z

The ZIP provider string determines which ZIP library is used for reading ZIP archives. The only supported value at this time is 7z (for 7-zip).

If this setting is not specified, the default library ICSharpZip is used.

### 6.3.21 [Job] INI-File Section

Set in this INI-File section the default values for the job options. See section [Job options](#) for available options.

### 6.3.22 [PdfOptimize] INI-File Section

To set options here, the 3-Heights® PDF Optimization Shell tool is required, see [Optimization](#) for more information.

#### Standard PDF optimization profiles

**Key:** archive

**Key:** print

**Key:** web

Three predefined PDF Optimization Shell profiles.

#### custom

**Key:** custom

Set a custom profile for the PDF Optimization Shell tool.

#### exepath

**Key:** exepath Default: pdfoptimize.exe

Specify the system path to the PDF Optimization tool executable (specify the absolute path if the pdfoptimize.exe is not on the local PATH variable).

**Note:** The 3-Heights® PDF Optimizer Shell tool requires a separate installation and license configuration.

## 6.3.23 [Options] INI-File Section

### AllowedTransforms

**Key:** `AllowedTransforms` Default:

List of installed transformations that may be used. This value shall consist of the command names and VBS script names that shall be available. See section [Document transformations](#) for detailed information.

Each name must be terminated by a colon.

### MAXTRANSFORMTIME

**Key:** `MAXTRANSFORMTIME` Default: `30`

Timeout setting in seconds for aborting a transform command.

### ForceClose

**Key:** `ForceClose` Default: `true`

When option is not defined, `false` will be used.

**true** Force termination of office application on termination.

**false** Do not force termination of office application on termination.

### plugins

**Key:** `plugins` Default:

List of plugin DLLs to be loaded. The DLL name can be enclosed by quote characters. Multiple DLL names are separated by blank spaces.

### TerminateApps

**Key:** `TerminateApps` Default: `true`

Terminate office applications on termination of the worker control program.

## 6.3.24 [ACRORD] INI-File Section

### MaxCallDuration

**Key:** MaxCallDuration Default: 120

Execution time limit for XFA rendering by Acrobat Reader. If this time limit is exceeded, the application is forcibly terminated.

## 6.3.25 [PDFOCR] INI-File Section

### MaxCallDuration

**Key:** MaxCallDuration Default: 600

Execution time limit for OCR recognition. If this time limit is exceeded, OCR processing is forcibly terminated.

## 6.4 Job options

### 6.4.1 ALIVECHECK

**Key:** ALIVECHECK

Setting this option tells the converter that the current conversion job is merely for verifying proper functioning. This has two effects:

1. The conversion does not contribute to the overall page count.
2. The output has an evaluation watermark.

**Note:** If OCR is performed, pages **do** count towards the monthly FineReader engine licensed limit.

### 6.4.2 CERTNAME

**Key:** CERTNAME

The certificate name to be used for signing the converted PDF. A certificate with the specified name must be installed in the certificate store, and be available to the Document Converter.

The **Name** corresponds to the common name (CN) of the subject. In the Windows' certificate store this corresponds to "Issued to".

To sign the output PDF with only a timestamp, use the **TS:** prefix for the name value. If a visible signature is created (see [SIGRECT](#)), the name is displayed without the prefix.

### 6.4.3 COLLCOMP

**Key:** COLLCOMP Default: true

Collect documents from composite items (such as email) into a single output document.

### 6.4.4 CONVERTALWAYS

**Key:** CONVERTALWAYS Default: true

Force PDF/A conversion. Only applicable when PDF to PDF/A conversion is performed.

### 6.4.5 EMBEDSOURCE

**Key:** EMBEDSOURCE

When converting to PDF, embed the source document as a file attachment into the resulting PDF document.

Specify the description text for the embedded file as value for this option. If an empty value or **true** is specified, "Original document" will be used.

(Not available with PDF/A-1 or A-2).

### 6.4.6 ERRSUMMARY

**Key:** ERRSUMMARY Default: false

**true** Set this option to collect all error messages and return them at the end of the conversion process. In particular this is useful when converting compound documents or using the web service.

### 6.4.7 ERRPAGE

**Key:** ERRPAGE Default: false

**true** Appends a page to the converted document containing an error summary.

### 6.4.8 FlattenSignatures

**Key:** FlattenSignatures Default: false

Keep signature appearance when removing digital signatures during conversion.

## 6.4.9 FlattenFormFields

**Key:** `FlattenFormFields` Default: `false`

Flatten form fields during conversion.

## 6.4.10 FORMAT

**Key:** `FORMAT`

Values `pdf` or `tiff`. Specifies the conversion output format.

For detailed configuration issues concerning TIFF conversion see section [TIFF output format](#).

## 6.4.11 HTMLPRINTRESOLUTION

**Key:** `HTMLPRINTRESOLUTION` Default: `1.0`

Specify a factor, relative to a screen resolution, for converting web pages.

## 6.4.12 LINEARIZE

**Key:** `LINEARIZE`

When `true`, optimize PDF output for fast web viewing.

## 6.4.13 MAILHEADER

**Key:** `MAILHEADER`

**false** Do not include headers in conversion.

**true** Add full mail headers to converted message.

**attach** Extract original headers and attach as text file.

The layout and styles used to produce the intermediate HTML representation of the mail message can be configured via a style sheet file and a header template file (see [Em1](#) INI-file section).

## 6.4.14 OCR

**Key:** `OCR`

Enable OCR recognition (values `true` and `false`).

**Note:** The ABBYY FineReader OCR engine must be installed and licensed separately to make this feature available.

**Note:** Use of [OCR.ENGINE](#) or [OCR.LANGUAGE](#) automatically turns this setting on.

## 6.4.15 OCR.BITONAL

**Key:** `OCR.BITONAL` Default: `false`

Pass color images converted to bitonal (black/white) image for OCR recognition.

This reduces the resources and time required for OCR recognition at the possible cost of somewhat reduced accuracy.

## 6.4.16 OCR.EMBEDBARCODES

**Key:** `OCR.EMBEDBARCODES` Default: `false`

**true** Embed barcode information into the document XMP metadata

For an example see chapter [OCR](#).

## 6.4.17 OCR.ENGINE

**Key:** `OCR.ENGINE`

Name of the OCR engine to be used for OCR processing (default: `service`).

For an example see chapter [OCR](#).

## 6.4.18 OCR.LANGUAGE

**Key:** `OCR.LANGUAGE`

Set the language to improve OCR recognition accuracy (values according to the engine; ABBYY expects English spelling, e.g. `German`, `English`).

For an example see chapter [OCR](#).

## 6.4.19 OCR.IMAGEMODE

**Key:** `OCR.IMAGEMODE`

Set the image mode for OCR recognition.

Available values are:

**none** Do not process images.

**update (default)** Only process images that have no OCR text.

**replace** Process all images and remove existing OCR text.

**remove** Remove existing OCR text.

**ifNoText** Process images only if document contains no text.

For an example see chapter [OCR](#).

## 6.4.20 OCR.TEXTMODE

**Key:** `OCR.TEXTMODE`

Set the text mode for OCR recognition.

Available option values are:

**none (default)** Do not process text.

**update** Only process text that is not extractable.

For all characters that have no meaningful Unicode, OCR processing is used to determine the Unicode. This is the recommended mode to make text extractable.

Note that making text extractable requires many OCR operations. The reason is that of all characters multiple instances must be recognized, to deal with erroneous OCR recognitions.

**replace** Process all text.

OCR is used to determine the Unicode of all characters, that is even if they seemingly have Unicode information. This is useful for documents that possibly contain wrong Unicode information. Wrong Unicode information is typically created by flawed PDF creators or to obfuscate text (i.e. to prevent copy-and-paste or search operations).

For documents that contain correct Unicode information, this mode produces the same result as the mode Update. The rare exceptions are special fonts for which the OCR engine produces wrong results, which might happen for some decorative or handwritten fonts. The main disadvantage of the mode Replace over Update is, that more OCR operations are required.

For an example see chapter [OCR](#).

## 6.4.21 OCR.PAGEMODE

**Key:** `OCR.PAGEMODE`

Set the page mode for OCR recognition.

Available option values are:

**none (default)** Do not process pages.

**all** Process all pages that are not empty.

**ifNoText** Process all pages that contain content but no text.

**addResults** Do not trigger processing of pages. But if pages are OCR processed, e.g. due to another OCR mode, add results as OCR text to pages.

For an example see chapter [OCR](#).

## 6.4.22 OCR.PARAMETERS

**Key:** `OCR.PARAMETERS`

Set OCR engine parameters.

For an example, see [OCR](#).

For detailed information, see OCR engine documentation.

## 6.4.23 OCR.REEMBEDIIMAGE

**Key:** `OCR.REEMBEDIIMAGE` Default: `false`

Re-embed images that have been processed (deskewed) during OCR recognition.

## 6.4.24 OCR.ROTATEPAGE

**Key:** `OCR.ROTATEPAGE` Default: `false`

If `true`, automatically rotate pages during OCR processing to have horizontal reading direction.

This setting only has an effect, if the underlying OCR engine is able and configured to detect the text orientation.

### Abby FineReader 11 or 12

The following profile configuration `abby_text.ini` needs to be configured to detect the text orientation:

```
[PagePreprocessingParams]
CorrectOrientation=TRUE
```

Use the profile using the OCR engine running on the OCR Service:

```
OCR.ROTATEPAGE=true;OCR.PARAMETERS=Profile="C:\Profiles\abby_text.ini
```

**Note:** It is important that `C:\Profiles\abby_text.ini` is the path to the configuration file on the OCR Service and the OCR Service process has read permissions.

## 6.4.25 OCR.TAGGING

**Key:** `OCR.TAGGING` Default: `false`

“Tagging” adds structural information to a PDF. This information can be used e.g. to read the document to the visually impaired.

This option controls, if this detected “tagging” information is generated for OCR text.

**true** Force embedding of tagging information. A warning is generated, if no tagging information can be added. Therefore, this value is recommended if tagging information is crucial to your process.

**false** Automatically detect whether tagging information should be added or not. Add tagging for scans and born-digital documents with tagging, and not otherwise.

## 6.4.26 PASSTHROUGH

**Key:** PASSTHROUGH

Pass source document without conversion to output (ZIPPED or PDF attachment). When converting to PDF, make sure to have at least one document converted.

**attachments (all mail attachments)**

**.xml .txt hidden none** List of Extensions.

**!.doc.eml.msg.zip** Negative list of documents with these extensions.

## 6.4.27 PDFA

**Key:** PDFA

Values **true** or **false**. When **true** is set, the resulting PDF will conform to the PDF/A standard.

## 6.4.28 PDFA.ERROR

**Key:** PDFA.ERROR

Values **true** or **false**. When **true** is set, return detailed error information from PDF/A conversion.

## 6.4.29 PDFA.LOGDETAILS

**Key:** PDFA.LOGDETAILS

Values **true** or **false**. When **true** is set, include detail information in error text.

Default: [PDFA.ERROR](#) setting.

## 6.4.30 PDFA.LOGSUMMARY

**Key:** PDFA.LOGSUMMARY

Values **true** or **false**. When **true** is set, include summary information in error text.

Default: [PDFA.ERROR](#) setting

## 6.4.31 PDFA.OCRMODE

**Key:** PDFA.OCRMODE

Deprecated as of version 5.5 - replaced by [OCR.IMAGEMODE](#)

Set one of the OCR modes applicable in the context of PDF/A conversion.

**1** OCR images without OCR text (default)

- 2 Remove old OCR text (and perform OCR).
- 3 Remove old OCR text (but do not OCR).
- 4 OCR if the document contains no text.

### 6.4.32 PDFA.EMBEDALLFONTS

**Key:** PDFA.EMBEDALLFONTS Default: false

By default, fonts are not embedded unless required to achieve PDF/A conformance. For example, fonts of OCR text (invisible text not used for rendering) must not be embedded. Setting PDFA.EMBEDALLFONTS to true forces all fonts to be embedded. This produces larger PDF/A output files and is intended only as a workaround for bugs in subsequent systems.

### 6.4.33 PDFA.SUBSET

**Key:** PDFA.SUBSET Default: true

By default, fonts that are embedded are automatically subset to minimize the file size. If for any reason, e.g. post-processing, fonts shall not be subset, set the property PDFA.SUBSET to false. Whether fonts are subset or not is irrelevant with respect to the conformance to PDF/A. (Relevant is only that all used glyphs are contained in the font program.) Setting this option to false will produce much larger PDF/A output files; however, these files may be easier to enhance or modify later on.

**Warning:** This setting is strongly discouraged. If this property is set to false, embedded fonts, that are subsetted, are replaced with non-subsetted fonts from the local system. To avoid visual differences, embedded fonts are only replaced if they match the system font. Therefore setting PDFA.SUBSET to false is not recommended, unless it can be guaranteed that all fonts match.

### 6.4.34 PDFA.WARNCOLL

**Key:** PDFA.WARNCOLL Default: true

Warn if an input PDF contains embedded files that need to be removed for PDF/A-1 conformance.

### 6.4.35 PDFA.WARNDOWNGRADE

**Key:** PDFA.WARNDOWNGRADE Default: false

Returns the warning O2P\_W\_DOWNGRADE (see [Error codes](#) in case the conformance is downgraded during PDF/A conversion).

## 6.4.36 PDFA.WARNUPGRADE

**Key:** PDFA.WARNUPGRADE Default: **false**

Returns the warning O2P\_W\_UPGRADE (see [Error codes](#) in case PDF/A conformance level was upgraded from PDF/A-1 to PDF/A-2.

## 6.4.37 PDFA.WARNNOTPDFA

**Key:** PDFA.WARNNOTPDFA Default: **false**

Returns the warning O2P\_W\_NOTPDFA (see [Error codes](#) in case PDF/A conversion failed and the (plain) PDF is returned.

## 6.4.38 PDFA.WARNFONTSUBST

**Key:** PDFA.WARNFONTSUBST Default: **false**

**true** Warn if a Font is substituted.

## 6.4.39 PDFA.WARNVISDIFF

**Key:** PDFA.WARNVISDIFF Default: **true**

**false** Suppress the warning normally raised in case PDF/A conversion possibly results in visual changes.

## 6.4.40 PDFA.XMPWARNINGS

**Key:** PDFA.XMPWARNINGS Default: **false**

**true** Raise a conversion error if the XMP Metadata was changed in order to achieve PDF/A conformance.

## 6.4.41 PDFA.CONVERTEMBPDF

**Key:** PDFA.CONVERTEMBPDF

**true** Embedded PDF files will also be converted to PDF/A.

**false** Embedded PDF files remain as-is.

If the option is not specified, the converter assumes **true** with PDF/A-2 and **false** with PDF/A-3.

## 6.4.42 PDF.COMPLIANCE

**Key:** PDF.COMPLIANCE Default: **2AUB**

Required (minimum) conformance level (for PDF/A conversion).

When option is not defined, **1AB** will be used.

A conformance is defined by values: **1A, 1B, 2A, 2B, 2U, 3A, 3U, 3B**.

Additionally a fallback value can be defined. So if you prefer **1A**, but will also accept **1B** in cases where **1A** is not possible, you can define **1AB**.

**1A** Raise an error if tagging information is missing.

**1B** Produce PDF/A-1b output, even if structure tags are available.

**1AB** Try to create PDF/A-1a; degrade to PDF/A-1b if tagging information is missing.

**2A** Produce PDF/A-2a (or fail).

**2UB** Produce PDF/A-2u or – if text encoding information is missing – PDF/A-2b.

**2AUB** Produce the “best possible” PDF/A-2 output

**3AUB** Produce the “best possible” PDF/A-3 output

**1AB, 2AUB** Prefer conversion to PDF/A-1, allowing upgrade to PDF/A-2 (e.g. due to transparency being used)

**2AUB, PDF** Attempt conversion to PDF/A-2, allowing plain PDF (e.g. when a font to be embedded is missing)

### 6.4.43 PDF.DATE

**Key:** PDF.DATE

The calendar date and time to be used for the output PDF. This will set the **CreationDate** entry in the document's Info object.

Values specified for this key must be formatted as this: **20071031141000** (October 31, 2007, 2:10 p.m. local time), or **20071031141000+ '02:00'** (same date, explicit UTC offset +2 hours)

### 6.4.44 PDF.Embed

**Key:** PDF.Embed Default: **false**

**true** When converting multiple documents, embed any but the first document into the resulting PDF document as document level attachments.

**false** Merges the pages of all documents.

### 6.4.45 PDF.Info

**Key:** PDF.Info

Document level attributes like Author, Title, etc.

E.g. **Author:Document Converter|Keywords:pdf-tools.com**

### 6.4.46 PDF.InitialView

**Key:** PDF.InitialView

Set preferences how a viewer should initially present the PDF document. The document converter supports ViewerPreferences, PageLayout and PageMode settings according to the PDF specifications (ISO 32000-1).

Several settings can be combined as shown in the example below. Individual settings can be separated with a comma or a semicolon. The comma will not be recognized as a separator when the value consists of numbers (in the case of PrintPageRange).

Boolean values need only be specified when "false" (but can be set to "true") - see examples.

#### Example:

```
PDF.InitialView=PageLayout:TwoColumnLeft,PageMode:UseOutlines,CenterWindow,DisplayDocTitle
```

```
PDF.InitialView=PageLayout:SinglePage,CenterWindow:false
```

```
PDF.InitialView=PrintPageRange=1,4,9,10;NumCopies:2;Duplex
```

### 6.4.47 PDF.OWNERPASS

**Key:** PDF.OWNERPASS

The owner password for creating a password protected output PDF (The password required to modify the document security settings). Be aware that PDF.USERPASS must be set as well, but can stay empty (no password required).

#### Example:

```
PDF.OWNERPASS=password;PDF.USERPASS;PDF.PERMISSION=0
```

**Note:** Do not use this for PDF/A conforming output.

### 6.4.48 PDF.PERMISSION

**Key:** PDF.PERMISSION

What operations in a PDF document are granted is controlled via its permission flags.

In order to set permission flags, the PDF document must be encrypted and have an owner password. The owner password is required to initially set or later change the permission flags

The permissions that can be granted are listed below.

### Permission flags

Flag	Description
p	allow printing (low resolution)
m	allow changing the document
c	allow content copying or extraction
o	allow commenting
f	allow filling of form fields
s	allow content extraction for accessibility
a	allow document assembly
d	allow high quality printing
∅	allow nothing (no permissions are granted)

For further information about the permission flags, see PDF Reference 1.7 Section 3.5.2.

**Note:** Do not use this for PDF/A conforming output.

## 6.4.49 PDF.Producer

**Key:** PDF.Producer

This option takes a string as value and sets it as the producer entry in the metadata of the converted document.

## 6.4.50 PDF.USERPASS

**Key:** PDF.USERPASS

The user password for creating a password protected output PDF. (The password is required to open the document).

**Note:** Do not use this for PDF/A conforming output.

## 6.4.51 PDFOPTIMIZE

**Key:** PDFOPTIMIZE Type: Enterprise

Perform a predefined command on the output PDF prior to the PDF/A conversion (or just signing) stage.

For details, refer to section [Optimization](#).

## 6.4.52 SIGEMBEDOCSP

**Key:** SIGEMBEDOCSP Default: true

Embed 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 certificate 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.

The downsides of embedding revocation information are the increase of the file size (normally by around 20k) and that it requires a connection to a validation service, which delays the process of signing (normally by around 2 seconds).

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-response

## 6.4.53 SIGFIELDS

**Key:** SIGFIELDS Default: false

If true, replace signature placeholders by signature fields that can later be signed (e.g. with Adobe Acrobat), see the [Placeholder image](#).

## 6.4.54 SIGISSUER

**Key:** SIGISSUER

Certificate issuer: This value is optional; it is only needed if multiple certificates with the same name exist.

Issuer corresponds to the common name (CN) of the issuer. In the Windows' certificate store this corresponds to "Issued by".

## 6.4.55 SIGPAGE

**Key:** SIGPAGE Default: -1

The page number on which to place the signature (default: -1 for last page).

## 6.4.56 SIGPROFILE

**Key:** SIGPROFILE

Name of a signature configuration file having INI file Syntax with sections [Session] and [Properties]. The name path can be relative to the installation folder and may have the .ini extension omitted.

[Session] Section keys are: Provider, Name, Reason, Text1, Text2, TimeStampUrl, TimeStampCredentials, ProxyURL, ProxyCredentials, EmbedRevocationInfo, SignerFingerPrint, SerialNumber, ContactInfo, Email, Location, ImageFileName.

[Properties] Section keys are specific for the particular provider value. Please refer to the appropriate documentation.

## 6.4.57 SIGPROVIDER

**Key:** SIGPROVIDER

Signature provider string: to use a PKCS#11 provider, specify the path to the PKCS11 interface DLL, followed by a semicolon, then the slot number, then another semicolon, and then the user PIN.

Additional information about PKCS#11 and HSMs is available in the [Tech-Note for HSM](#).

## 6.4.58 SIGPROXYCRED

**Key:** SIGPROXYCRED

Proxy credentials (user:password).

## 6.4.59 SIGPROXYURL

**Key:** SIGPROXYURL

Proxy URL (for accessing timestamp and OCSP server).

## 6.4.60 SIGREASON

**Key:** SIGREASON

The signature reason.

## 6.4.61 SIGRECT

**Key:** SIGRECT

The signature rectangle (4 space separated float values: left bottom right top).

For invisible signatures, specify a rectangle with zero width and height.

## 6.4.62 SIGSTORE

**Key:** SIGSTORE

Certificate store (**MY**, **ROOT**, **CA**, ...).

### 6.4.63 SIGSTORELOCATION

**Key:** SIGSTORELOCATION

Certificate store location (**0** local system; **1** current user).

### 6.4.64 SIGTSCRED

**Key:** SIGTSCRED

Timestamp server credentials if required (**user:password**).

### 6.4.65 SIGTSURL

**Key:** SIGTSURL

The URL of the trusted timestamp authority (TSA) from which a timestamp is acquired. E.g. **tsu.my-time-server.org**.

This setting is suggested to be used when applying a Qualified Electronic Signature. Applying a timestamp 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

### 6.4.66 WARNSIGNATURES

**Key:** WARNSIGNATURES Default: **false**

**true** Warn if an input PDF files is digitally signed.

### 6.4.67 STAMP

**Key:** STAMP

Name of an XML stamp file. See [Stamping](#).

**Note:** A suitable domain account must be specified for the watched folder service in the "login" tab to ensure that files on a network share can be accessed. The watched folder service will move the control file to the succeeded folder (or deleted it) on job completion. The documents referenced in the control file are left untouched.

## 6.4.68 TEXTBODY

**Key:** TEXTBODY Default: false

**true** Include the text body in the conversion of mail messages also, if HTML or RTF body is present too.

## 6.4.69 THUMBS

**Key:** THUMBS

Values **true** or **false**; using the option **THUMBS=true** will create thumbnails for all pages. Existing thumbnails will be replaced. **THUMBS=false** will have no effect on thumbnails; existing thumbnails in input documents will not be altered.

## 6.4.70 TIFF.BPI

**Key:** TIFF.BPI

Specify the Bits per pixel of a TIFF output.

## 6.4.71 TIFF.COMPR

**Key:** TIFF.COMPR

Set the TIFF compression type. Possible values are **raw**, **JPEG**, **flate**, **LZW**, **Group3**, **Group3\_2D** and **Group4**.

## 6.4.72 TIFF.COMPR.BITON

**Key:** TIFF.COMPR.BITON Default: Group4

Set the TIFF compression type for bitonal images (black and white). Possible values are **raw**, **flate**, **LZW**, **Group3**, **Group3\_2D**, **Group4** and **PackBits**.

## 6.4.73 TIFF.COMPR.CONTINUOUS

**Key:** TIFF.COMPR.CONTINUOUS Default: TIFFJPEG

Set the TIFF compression type for continuous color images. Possible values are **raw**, **JPEG**, **flate**, **LZW** and **TIFFJPEG**.

## 6.4.74 TIFF.COMPR.INDEXED

**Key:** TIFF.COMPR.INDEXED Default: LZW

Set the TIFF compression type for color indexed images. Possible values are **flate**, **LZW** and **raw**.

## 6.4.75 TIFF.COMPR.MRC

**Key:** TIFF.COMPR.MRC

**true** Activate MRC.

## 6.4.76 TIFF.COMPR.PDF

**Key:** TIFF.COMPR.PDF

Set the TIFF compression type for PDF to TIFF. Possible values are **raw**, **JPEG**, **flate**, **LZW**, **Group4**, **PackBits** and **TIFFJPEG**

## 6.4.77 TIFF.DITHERINGMODE

**Key:** TIFF.DITHERINGMODE Default: **1**

The dither mode to be applied during PDF to TIFF conversion with bi-level compression.

- 1** (Floyd Steinberg) for JBIG2
- 4** (Group3 optimized) for Group3
- 5** (Group4 optimized) for Group4

## 6.4.78 TIFF.DPI

**Key:** TIFF.DPI

Set the target resolution in dots per inch. Applies to PDF to TIFF and Excel to TIFF.

## 6.4.79 TIFF.RENDERINGMODE

**Key:** TIFF.RENDERINGMODE Default: **1**

The rendering mode to be applied for PDF to TIFF conversion.

- 1** (accurate) for color conversions
- 0** (fast) for black and white (Group3, Group4, JBIG2 compression)

## 6.4.80 TIFF.ROTATE

**Key:** TIFF.ROTATE Default: **Auto**

Image rotation policy for PDF to TIFF conversion. Possible values are **Auto**, **None**, **Landscape** and **Portrait**.

## 6.4.81 TIFF.UNPACK

**Key:** `TIFF.UNPACK`

**true** Unpack samples to 8 bits per sample.

## 6.4.82 TIFF.XDPI, TIFF.YDPI

**Key:** `TIFF.XDPI` Default: `150`

**Key:** `TIFF.YDPI` Default: `150`

The horizontal and vertical resolution in dots per inch. To set the same resolution for both, you can use the option name [TIFF.DPI](#).

## 6.4.83 ZIPPED

**Key:** `ZIPPED`

Values **true** or **false**. When **true**, documents of a job are returned as multiple entries in a ZIP file rather than concatenated into a single TIFF or PDF file.

## 6.4.84 ZUGFeRD-invoice.xml:ADD

**Key:** `ZUGFeRD-invoice.xml:ADD`

**A** Add ZUGFeRD-invoice.xml files as attachment.

## 6.4.85 factur-x.xml:ADD

**Key:** `factur-x.xml:ADD`

**A** Add factur-x.xml files as attachment.

## 6.4.86 Special key

**Key:** `#`

This is a special key controlling the way how documents are passed by the O2PProxyAPI .DLL to the server.

By default, documents are passed “by value”, i.e. their content is sent via the API. When **specifying this key at the very beginning of the options string**, document files are not copied, but their file system path is sent.

Obviously, this will only work when the file system path has a meaning on the server side, and the server has access privileges to read the documents and also create the resulting PDF.

The advantage of using this key is that the copying overhead is eliminated.

## 6.5 Document options

Options related to the processing of a particular document.

It is possible to restrain options to the scope of a particular application or document by prefixing the option key with the application name or the document name, separated by a colon. This is useful when multiple documents are passed.

### Example:

```
HTML:Outline=TITLE;URL:Outline=FILENAME;Outlook:Outline=%Subject%
```

**Note:** Document options can also be used at the job level. If a certain option value is evaluated, the settings specified at the document level have priority over settings specified at the job level.

### 6.5.1 ADD

**Key:** ADD

Specify that the document be embedded as a document level attachment into the resulting PDF document (value **A**).

Optionally, the document can also be converted to PDF and merged with the output document (value **B**).

The setting of this option can be restricted to mail attachments. In this case, use the option key **ATTACHMENTS:ADD**.

### 6.5.2 ATT.Name

**Key:** ATT.Name

Specify the name to be used as the attachment name.

Using this document option implies **ADD=A**, unless **ADD=B** is specified explicitly.

The name **ZUGFeRD-invoice.xml** has a special meaning (also when specified via **ORIGINALNAME**). If an XML file with this name is passed, the converter service will propagate ZUGFeRD metadata from the XML document to the XMP metadata and embed the XML itself as an embedding (according to the ZUGFeRD specification).

### 6.5.3 ATT.Description

**Key:** ATT.Description

Specify a description string for the document to attach.

### 6.5.4 CMRQUAL

**Key:** CMRQUAL Default: 95

Compression quality (1, ..., 100) to be applied for lossy (re-)compression during PDF to PDF/A conversion and image to PDF compression.

## 6.5.5 CODEPAGE

**Key:** CODEPAGE Default: CP\_ACP

Set the code page for converting ASCII text files using MS Word. Default is CP\_ACP (currently set default Windows ANSI code page).

To disable code page mapping and pass ASCII text files "as is" to MS Word, specify a value of `false`.

## 6.5.6 COLLCOMP

**Key:** COLLCOMP

If set to `true`, the elements of a composite document are merged (collected) into a single output document (default `false` when `ZIPPED=true` is specified for the job's output).

Composite documents are Mail messages containing attachments, ZIP and RAR files.

## 6.5.7 FAILFILES

**Key:** FAILFILES

Cause the conversion of files having an extension listed in the FAILFILES setting to fail.

The special value `.unknown` can be specified to match any file extension that is not assigned to a known document type. Use this setting if you want to prevent the conversion of specific document types. This is more specific than just disabling the use of the corresponding application (e.g. Excel), as some other application (e.g. OpenOffice) may still attempt to perform the conversion.

## 6.5.8 FORCEAPP

**Key:** FORCEAPP

Use the specified value in place of the file name extension for controlling which application is selected to open the document.

If the value is omitted (or empty), the original file's extension will be used to select the controlling application; if that application cannot open the file, no other attempt to open the document with other applications will be made.

**Example:** Select the application registered for PPT (i.e. PowerPoint), and name the file `f.pptx` to open it.

```
ORIGINALNAME=f.pptx;FORCEAPP=PPT
```

**Note:** For certain file types and applications, it is important that the correct file extension is provided. Visio for example will not open any document with an inappropriate extension.

## 6.5.9 HTZDOCS

**Key:** HTZDOCS Default: .HTM.HTML.MHT.XML

The list of file extensions for selecting the root document(s) in .HTMZIP files.

## 6.5.10 Include Visio properties

**Key:** IncludeStructureTags

**Key:** IncludeDocumentProperties

**Key:** IncludeBackground

Create structure tags, include document properties or include the background layer when saving the document as PDF.

## 6.5.11 MAILPARTS

**Key:** MAILPARTS Default: AB

Select attachments and/or body of the mail message for conversion.

**A** Attachments only.

**B** Body only.

**L** All.

Use this option to determine which parts of a mail message shall be converted. Supported for Outlook MSG files and Internet Mail Messages (section [Conversion of Internet mail messages](#)).

## 6.5.12 ORIGINALNAME

**Key:** ORIGINALNAME

Specifies the original name of the document being passed.

The file name extension determines by which application(s) the file will be processed. Microsoft Visio for example will not even try to open a file if it does not have one of its known extensions.

The specified value is also the one that will be used when an outline (bookmark) title shall be set according to the document's file name.

## 6.5.13 AlternateText

**Key:** AlternateText

Specifies the alternate text for images converted to PDF. Use this for a target conformance level of A (e.g. PDF/A-2a).

Note: avoid specifying an alternate text for images that represent scanned text that will be OCR processed.

## 6.5.14 LanguageTag

**Key:** LanguageTag

Specifies the language identifier for images converted to PDF. Use this for a target conformance level of A (e.g. PDF/A-2a). If not specified, EN-US is used.

## 6.5.15 Outline

**Key:** Outline

Specifies that outlines (bookmarks) shall be created for the document and any documents contained within it.

If any contained documents are processed to have outlines, these outlines will only be copied if the parent document also has an outline to which these can be attached.

The following special values are supported to be replaced by document related information:

**FILENAME** Is replaced by the file name (without extension) of the current document.

**FILENAMEX** Is replaced by the file name and extension of the current document.

**%Subject%** Applicable for an Outlook mail or PDF document: supplies the subject field of the mail message.

Actually, it is possible to use other valid mail message properties, e.g. SenderName. %Subject% can be embedded in a plain text string.

**TITLE** Title entry of an HTML page on a web server.

**(empty)** An empty value has a special meaning: for an aggregate document, the outlines of the contained documents will be "lifted" a level and appear in the place of this document.

## 6.5.16 OutputType

**Key:** OutputType

Set the "OutputType" print option (PowerPoint). Default inherited from the service configuration (see [Application options](#) also for possible values and their meaning).

## 6.5.17 PRINTCOLOR

**Key:** PRINTCOLOR Default: false

Applicable when `FORMAT=tiff`; the default is `false` (convert to gray). Can also be specified as a document option.

## 6.5.18 PRINTDM

**Key:** `PRINTDM`

Values `A-1b`, `A-2u`, `1.7`, `1.6`, `1.5`, `1.4` (applicable for PDF conversions)

Values `draft`, `low`, `medium`, `high` (resolution; applicable for both PDF and TIFF output)

It is possible to combine different values by separating them with a comma, e.g. "1.6,medium".

Alternatively, the name of a device mode file located in the installation folder can be passed.

Using this option setting will prepare the device mode settings of the printer to be used by office applications during the conversion of a document.

## 6.5.19 PW

**Key:** `PW`

Password to open a password protected document (Word, Excel, PDF, ZIP, RAR).

**Note:** AES-256 decryption for ZIP files is only available through the "7z" provider (see [PROVIDER](#) setting).

## 6.5.20 SAVEASPDF

**Key:** `SAVEASPDF`

**true** Use the Office "Save As PDF" feature rather than the 3-Heights® PDF Producer.

## 6.5.21 RemoveVBA

**Key:** `RemoveVBA` Default: `true`

**true** Pre-process Office Documents to remove VBA code prior to any further processing.

Documents with the file name extensions DOC, DOT, XLS and XLT are processed in-place. In most cases, this is not a problem, as these files are temporary copies. When passing documents to the converter via file name reference, it is the application's responsibility to either disable VBA removal or provide a copy of the original document, if that shall not be modified (or is read-only).

Documents with the file name extensions PPTM, XLSM, XLSB, DOCX, DOCM, DOTM, XLTM and XLSX are pre-processed not only for VBA code removal, but also for fixing issues with predefined names in Excel sheets. The original file is not modified.

**false** Disable preprocessing as described above.

Pre-processing can be disabled selectively by prefixing the option key with the file name extension (e.g. `PPTM:RemoveVBA=false`).

## 6.5.22 AcroRdEnableJS

**Key:** `AcroRdEnableJS`

**false** perform XFA conversions via Acrobat Reader with Javascript disabled.

This is the default setting and is recommended to avoid undesired side effects.

**true** perform XFA conversions via Acrobat Reader with Javascript enabled

This option may be configured statically in the ACRORD section of the O2PWSC.ini file.

## 6.5.23 TIFF.COMPR.BITONAL

**Key:** `TIFF.COMPR.BITONAL` Default: `Group4`

Compression method for bitonal images.

**Values** `Group4, Flate, LZW, Group3, Group3_2D, JBIG2`

## 6.5.24 TIFF.COMPR.CONTINUOUS

**Key:** `TIFF.COMPR.CONTINUOUS` Default: `JPEG`

**Values** `JPEG, Flate, LZW, JPEG2000`

## 6.5.25 TIFF.COMPR.INDEXED

**Key:** `TIFF.COMPR.INDEXED`

Default Flate for PDF/A output, LZW otherwise.

## 6.5.26 TIFF.COMPR.MRC

**Key:** `TIFF.COMPR.MRC` Default: `false`

Set **true** to enable mixed raster content compression.

## 6.5.27 TIFF.UNPACK

**Key:** `TIFF.UNPACK` Default: `false`

Set **true** to convert indexed images to continuous, permitting downsampling.

## 6.5.28 TRANSFORM

**Key:** TRANSFORM

License: Enterprise

If set, a custom prepared transformation utility is invoked prior document conversion (and after pre-validation).

For details, refer to [Document transformations](#).

## 6.6 Watched folder configuration

O2PWFS.exe is the watched folder executable. It actually constitutes a client application of the Document Converter. It uses the client API DLL (O2PProxyAPI.DLL) that itself relays any documents passed for processing to the (possibly remote) service.

The O2PWFS.ini file contains the entries that configure the watched folders and their corresponding processing options. See below for a detailed description.

### 6.6.1 [O2PWFS] INI-File Section

**Example:**

```
[O2PWFS]
ServiceHost = localhost
ServicePort = 7981
AutoDelete = false
JobPrefix = false
Threads=1
Thread1=-w d:\DocConverter -j PDFa -b Outline -o "C:\Temp"
```

#### AutoDelete

**Key:** AutoDelete Default: false

Delete input files of successfully converted documents.

#### AutoDeleteAll

**Key:** AutoDeleteAll Default: false

Delete input files of successfully converted documents and of failed jobs.

## JobPrefix

**Key:** JobPrefix Default: false

Rename files to contain a unique job specific prefix.

## KeepTimeForFailed

**Key:** KeepTimeForFailed

The number of hours until files in failed folder are deleted.

If this key is unspecified or 0 files are never deleted.

## KeepTimeForSucceeded

**Key:** KeepTimeForSucceeded

The number of hours until files are deleted.

If this key is unspecified or 0 files are never deleted.

## LogLevel

**Key:** LogLevel Default: 1

- 1 log errors only
- 0 informational log

## LogPath

**Key:** LogPath

File path for log file.

Default: Document Converter's log directory.

## PollingInterval

**Key:** PollingInterval

Milliseconds between file searches.

## ServiceHost

**Key:** ServiceHost

The computer name or network address where the Document Converter is hosted; `localhost` is a common network name alias referring to the local computer.

## ServicePort

**Key:** ServicePort

ServicePort is the port number configured for the Converter Service (usually 7981).

## Threads

**Key:** Threads

The total number of threads or watched folders.

## Thread

**Key:** Thread

For each watched folder there is thread entry, starting with Thread1, Thread2 etc.

See subsection [Watched folder thread options](#) below for options which can be set as values of the `Thread1` key. Multiple options are set as a space separated list.

## WorkerThreads

**Key:** WorkerThreads

Maximum number of concurrent worker threads; use this to control the maximum load on the Document Converter.

# 6.7 Watched folder thread options

## 6.7.1 -b Set document options

**Set document options** -b <opt>

Define with the option **-b** all document conversion options specific for the watched folder.

Multiple parameters of **-b** are separated with a semicolon. To set a document conversion option relating only to a specific application (i.e. set in a specific INI-Section), use a colon to refer to this application as in the example below.

See section [Document options](#) for valid Parameters of **-b**.

**Note:** **-b** overrides the [Service configuration editor](#).

**Example:** Multiple job options.

```
-b Outline=FILENAME;Outlook:Outline=%Subject%;SAVEASPDF
```

## 6.7.2 -di Delete input files

**Delete input files** -di

Delete the input files listed in job control files

## 6.7.3 -j Set converter job options

**Set converter job options** -j <opt>

Define with the option **-j** all document processing options specific for the watched folder, multiple options for **-j** are separated with a semicolon. See section [Job options](#) for valid Parameters of **-j**.

**Note:** **-j** overrides the [Service configuration editor](#).

**Example:** Multiple job options.

```
-j PDFFA;PDFFA.ERROR;CONVERTALWAYS
```

## 6.7.4 -l Create error log

**Create error log** -l

Create an error log file in case of errors during the conversion process. The file will be located in the output directory.

## 6.7.5 -o Specify output directory

**Specify output directory** -o <dir>

This option can be used to specify the directory location for storing result PDF files (the default being the sub-folder PDFs or TIFFs in the watched folder directory specified with the option **-w**).

A folder relative to the pickup/drop directory is specified by prefixing the folder name with a colon (e.g. **:PDF/A**).

## 6.7.6 -of Specify failed directory

**Specify failed directory** -of <dir>

Specify the directory for storing input files that have failed (default: `Failed` folder in root specified with the option `-w`).

## 6.7.7 -op Unprefix output

**Unprefix output** -op

Remove job prefix of output file after conversion.

Job prefixes are useful to avoid name conflicts and to ensure proper sequencing of multiple conversion jobs.

## 6.7.8 -os Specify directory for succeeded jobs

**Specify directory for succeeded jobs** -os <dir>

Set the directory for succeeded jobs (default: `Succeeded` folder in root directory specified by `-w`). A folder relative to the pickup folder can also be specified by prefixing the folder name with a colon.

## 6.7.9 -ow Ignore warnings

**Ignore warnings** -ow

The option `-ow` applies to conversion jobs producing PDF/A conforming output (i.e. the job option `PDF/A` is also set). Whenever the PDF/A-conversion encounters a situation where the appearance of pages might change due to conversion processing, a warning is returned.

When the option `-ow` is set, such warnings are ignored, as long as the final output has been successfully validated as PDF/A).

**Note:** Use this option with caution, changes of page content are possible (e.g. removal of transparency).

## 6.7.10 -owf Warnings output folder

**Warnings output folder** -owf <dir>

Ignore any warnings (related to PDF/A conversion or parts that are not convertible), but move the result to a specified folder.

## 6.7.11 -ox IDX name

**IDX name** -ox

Specify a name pattern for the index file path to be used with option [-1](#) an [-11](#).

A relative path can be specified and will be based on the output folder for the conversion job. The placeholder \* will be substituted by the job's name.

### Example:

```
-ox C:\Completed-IDX\*.txt
```

## 6.7.12 -o0 Keep output

**Keep output** -o0

Conversion results are stored by the watched folder service as obtained from the converter service, but are deleted when an error or warning was encountered. Use the option `-o0` to keep the output document in case of a warning or error.

Actually, the output document will shortly appear in the output folder, but be deleted as soon as the job completes.

## 6.7.13 -o1 Second output directory

**Second output directory** -o1 <dir>

Store another copy of the output in a second folder specified as the argument to this option.

## 6.7.14 -o2 Force succeeded

**Force succeeded** -o2

Documents picked up in a drop folder will always be copied to the output folder. When a job fails, another copy of the input document will still be stored in the `Failed` folder.

## 6.7.15 -R Search for dropped files

**Search for dropped files** -R

Search in subfolders of drop directory also, and store output in corresponding subfolder of output directory.

## 6.7.16 -u Unzip output to folder

**Unzip output to folder** -u

Set the "ZIPPED" job setting (`-j ZIPPED=true`) and store each entry of the output archive in a subfolder of the output directory.

## 6.7.17 -w Specify the path to the root directory

```
Specify the path to the root directory -w <dir>
```

This option sets the path to the root directory.

**Note:** This parameter must always be the first parameter of a [Thread](#).

### Parameter:

<dir> The given path should not contain mapped drives, since other users (such as LocalSystem) do not recognize them.

### Example:

```
Thread1=-w C:\ConverterService\WorkFolder1
```

The service supports path lengths including file name of up to 258 characters. This includes the 21 characters of the job ticket.

If a file name exceeds this value, its file name is truncated at the end of the file name and before the file extension. Therefore, watched folder names should be kept reasonably short.

## 6.7.18 -wd Specify the drop path

```
Specify the drop path -wd <dir>
```

This option sets the path of the drop directory. If this option is not set, then the drop directory equals the root directory `-w`. The drop directory is the watched folder, where input files are picked up and processed by the service.

### Parameter:

<dir> The drop directory can be at any existing (network-) location with the following conditions:

- The directory exists.
- The user under which the service runs has access permissions to this directory.

### Example:

```
-wd C:\Path\DropFilesInHere
```

## 6.7.19 -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.

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

## 6.7.20 -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 JPEG and PNG files.

```
-wfs .jpeg.jpg.png
```

## 6.7.21 -1 Single page output

```
Single page output -1
```

Store TIFF result in a collection of single-page TIFF files.

In addition, a file with ending `.idx` is created that contains the names of the single page files.

## 6.7.22 -11 Single page output

```
Single page output -11
```

Similar to [-1](#), but store the full file path in the `.idx` file.

## 6.8 Dispatcher service - O2PSRV.exe.config

The application configuration file for `O2PSRV.exe` contains the entries that are needed to start up the Worker Session. Consider section [Architecture](#) in the appendix for more information.

**Example:**

```
<?xml version="1.0" encoding="utf-8" ?>  
<configuration>
```

```

<appSettings>
  <add key="ServicePort" value="7981"/>
  <add key="ServiceBindTo" value=""/>
  <add key="WorkerCount" value="1"/>
  <add key="WorkerPath" value="C:\ConverterService\O2PWSC.exe"/>
  <add key="WorkerServer" value="localhost"/>
  <add key="WorkerUserName" value="pdf-tools\service"/>
  <add key="WorkerPassWord" value="*****"/>
  <add key="TempDirectory" value="C:\ConverterService\temp"/>
  <add key="CleanupTemp" value="true"/>
  <add key="LogFile" value="C:\ConverterService\log\o2psrv-DATE.log"/>
  <add key="LogLevel" value="2"/>
  <add key="WorkerLogFile" value="C:\ConverterService\log\o2psrv-w-DATE.log"/>
  <add key="WorkerLogLevel" value="2"/>
  <add key="StartupDelay" value="30"/>
</appSettings>
</configuration>

```

The entries in the `appSettings` scope of the configuration file are explained in more detail in the following.

## appSettings

**ServicePort** The TCP port number on which the service publishes its server interface.

**7981** Default.

**ServiceBindTo** Bind the service remote channel to a specific network interface, designated by the IP address.

**""** Default: accept connections from all interfaces.

**WorkerCount** The number of worker sessions to be started by the service.

**1** At least one worker is needed; two or more permit higher performance and scalability.

**WorkerPath** The path of the worker session control executable. This must be a local directory on the computer named by the `WorkerServer` entry.

**C:\Bin\O2PWSC.exe**

Default: Document Converter Service directory.

**WorkerServer** The network name of the computer on which the worker session shall execute.

If multiple worker sessions shall execute on distinct computers, use `WorkerSession1`, `WorkerSession2` etc. to specify multiple values.

**localhost** To specify a custom RDP port, append a colon and the port number (e.g. `localhost:3388`)

**WorkerUserName** The account name under which the worker session shall be started.

**service** Default.

**WorkerPassWord** The login password for the worker session account.

**""** Default.

This password has to be mangled, so it cannot be quickly recognized or directly be used. Mangling is performed by replacing each character of the password by its successor in the ASCII table. E.g. for `ABC0` specify `BCD1`.

**MaintenanceDuration** Duration in hours until workers logged off for maintenance will be started again.

**1** Format: float (e.g. **1.5** for 90 minutes).

**Note:** Workers are automatically restarted when conversion requests are made.

**MaintenanceHour** Logoff idle workers at specified time of day.

This is to permit Windows Update to automatically perform a server reboot (unless other users are still logged in interactively). To disable this feature, specify an empty value.

If the document converter is expected to be permanently available, disable maintenance mode. A regular restart of the service (or the entire server) however is recommended, for example during low use periods on weekends.

**3** Default.

The value is a float number, e.g. **3.5** for 3:30 in the morning.

**RestartHours** A comma separated list of the daily hours (local time) as decimal numbers when worker sessions shall be restarted. This key should only be used with the worker session instance number appended to avoid service degradation due to simultaneous restart of all sessions.

"" Default.

**Example:**

```
<add key="RestartHours1" value="12.5"/>
```

**RestartPeriod** The maximum time period until restarting a worker (in seconds).

Use of this setting for production environments is discouraged.

"" Default.

**SingleOutlook** Dispatch Outlook related documents to a single dedicated worker instance per WorkerServer/WorkerUserName pair. The Outlook application can be opened only once by a specific user on a terminal server.

**false** Default.

**DisconnectUsers** Use this setting to control which active RDP connections can be disconnected in case of need to start worker sessions. If not configured, any user but the user 'Administrator' are subject to be disconnected. This is equivalent to the value **!administrator,\***. The value is a comma separated list of user names. An exclamation point before the user name means that this user may not be disconnected, and the asterisk matches all names. Names are not case sensitive.

**DisconnectMaxActive** Use this setting to change the default behavior for controlling RDP session disconnections as described above. If the server runs in administrative mode, the default value is 1. This means that no users will be disconnected as long as one or no RDP connection is active. If the server has Remote Desktop Services role installed, more than two simultaneous RDP connections can be established on the server. In this case, a value of 99 is assumed (i.e. the document converter assumes that 100 or more RDP connections can be active simultaneously).

**TempDirectory** The absolute path to an existing directory that is accessible for both the dispatcher service and the worker session. A UNC path can be specified, if these programs execute on different computers.

If this entry is missing, the system default temporary directory is used.

**CleanupTemp** Periodically clean up left-over files in the TempDirectory folder.

**true** Default.

However if the key **CleanupTemp** is missing, the value is **false**.

**DisableLogonMessage** Temporarily disable the interactive logon message configured on the server during starting worker sessions.

**true** Default.

However if the key **DisableLogonMessage** is missing, the value is **false**.

**RunDisconnected** Automatically disconnect the worker session once it is established.

This is useful on Server 2008 platforms, as disconnected sessions do not count towards license based limits.

**true** Default.

However if the key **RunDisconnected** is missing, the value is **false**.

**LogFile** The absolute path of the log file to be written.

**[LOGDIR]\o2psrv-DATE.log**

The Document Converter's log directory is by default created as subfolder of the installation folder.

The log file name for the dispatcher log is either o2psrv.log or o2psrv-DATE.log, which DATE being a placeholder for the current day's date (4 year digits, 2 digits each for month and day). It is written by the dispatcher service and contains high level logging information about (re-)starting and stopping workers, managing conversion jobs, and dispatching documents to workers.

The service also performs processing tasks such as PDF merging, PDF/A conversion and OCR, and log this to the worker log file which is usually named as o2psrv-w.log or o2psrv-w-DATE.log (unless the [WorkerLogFile](#) key is defined explicitly).

**LogLevel** The log level controls filtering of log messages.

**2** Default.

Debug=**1**, Info=**2**, Error=**3**

**SvcLogLevel** The log level controlling service log entries into the system event log.

**3** Default.

**WorkerLogLevel** Set the log level for PDF and raster image processing work performed in-process by the dispatcher service.

**(none)** Default: no logging

**WorkerLogFile** File containing log entries from PDF and raster image processing work, including OCR. These entries are more detailed than the log entries generated in the standard o2psrv.log.

**C:\ConverterService\o2psrv-w.log**

Default: Document Converter's log directory.

**LogUTC** Print date/time information in UTC rather than local time.

**false** Default.

**StartupDelay** Time span in seconds to wait after starting the service before attempting to start the worker sessions.

Use of this feature helps to avoid session startup problems after booting due to system overload.

**30** Default.

**Note:** Specifying `node1ay` as the first parameter when starting the service will disable the configured `StartupDelay`.

**WorkersBusyTimeout** Timeout period to wait for when all worker sessions are busy before returning an error.

**30** Default.

**WorkerConcurrency** Limit for the number of concurrent conversions dispatched to any worker session.

**Note:** Increasing this number can result in stability problems with Microsoft Office applications. The recommended way to increase concurrency for the conversion of office documents is to increase the number of worker sessions (see [WorkerCount](#)).

**1** Default.

**WorkerTimeLimit** Time limit in seconds for performing the conversion of a document by a worker. The service will forcibly logoff the worker's session when reaching this limit (and then restart a new session).

**600** Default.

**Note:** When setting the `TempDirectory` and `LogFile` values, consider the security settings for the account under which the Dispatcher Service is running. Depending on the account under which the service runs, it may not have any write privilege unless you explicitly grant it.

**Worker instance suffix** Each of the worker session related settings can be individually configured by adding the instance suffix to the corresponding key.

Examples: `WorkerUserName1`, `WorkerPassword1`, `RestartHours1`, `WorkerServer1`, etc.

## 6.9 Worker session control - O2PWSC.exe.config

Consider section [Architecture](#) in the appendix for more information.

This section lists the configuration entries that apply to the worker session control program `O2PWSC.exe`

### appSettings

**ServiceHost** The computer name or network address for binding to the dispatcher service.

**localhost** Default.

**ServicePort** The port number configured for the dispatcher service

(See [Dispatcher service - O2PSRV.exe.config](#))

**7981** Default.

**DispatcherAliveCheck** Time interval in seconds for calling the dispatchers `AreYouStillAlive` method.

**60** Default.

Specify the value `false` to disable this feature.

**LogFile** The absolute path of the log file to be written.

**C:\o2pWSC-%ID-%Y%M%D.log**

Default: Document Converter's log directory.

`%ID` is a place holder for the instance number of the worker session.

`%Y`, `%M` and `%D` are place holders for the current date, i.e. year, month and day.

**LogLevel** This value controls the level of details to be logged.

**1** debug + below

**2** informational + below (Default)

**3** warnings + errors

**4** errors only

**CleanRecentInterval** The number of seconds between cleaning of user's Microsoft\Windows\Recent and Microsoft\Office\Recent folders.

**20** Default.

**0** or a smaller value to disables this feature.

## 6.10 Error codes

The 3-Heights® Document Converter specific errors are listed in the following table.

**Error codes**

Error Code and Message	Background
O2P_E_UNKFORMAT = 0x82410C01 The file has an unknown format	The input document format could not be recognized. The document may be corrupted.
O2P_E_INVALIDOP = 0x82410C02 Invalid state for requested operation	An inappropriate API call was made (e.g. wrong call sequence).
O2P_E_NOPAGES = 0x82410C03 The document contains no pages	The document to be converted does not contain any pages.
O2P_E_NOPDFPRINTER = 0x82410C04 No suitable PDF Printer installed	The PDF Producer printer entries are not available or have been deleted.
O2P_E_NOSCREENS = 0x82410C05 No screen session available	Terminal server sessions are not available.
O2P_E_PRINTTIMEOUT = 0x82410C06 Printing timeout experienced	The output from printing is overdue. The log file may contain more information.

## Error codes

<code>O2P_E_SVCUNAVAIL = 0x82410C07</code> Document Converter unavailable	The client failed to connect to the (remote) server. It may not have been started yet, or the URI may not be configured correctly.
<code>O2P_W_PARTSMISSING = 0x02410C08</code> Some parts of the document could not be processed	A compound document (mail with attachments, ZIP archive) contains elements that could not be converted.
<code>O2P_E_PDFACONVFAIL = 0x82410C09</code> PDF/A conversion failed	Failure may be due to transparency in the input document, missing color profiles, or other issues. See log files.
<code>O2P_E_UNKNOWN = 0x82410C0A</code> Generic error	The log file may contain more information.
<code>O2P_E_APPERROR = 0x82410C0B</code> Print application specific error	MS Word or some other third party application has encountered a problem. See log files.
<code>O2P_W_PDFACONVWARN = 0x02410C0C</code> PDF/A conversion completed with warnings	See log files or error description returned for more detailed information.
<code>O2P_E_SOURCECORRUPT = 0x82410C0D</code> Source document validation failed	An input document is not conforming to validation criteria. To configure MS Office Binary File Format validation, use <a href="#">BFFValidate</a> (Word, Excel, PowerPoint). Image document validation is controlled via the <a href="#">FailSize</a> option.
<code>O2P_W_SOURCEQUALITY = 0x02410C0D</code> Source document inconsistencies detected	An input document contains invalid links or bookmarks or is not a conform Office document. To configure document validation, use <a href="#">BFFValidate</a> (Word, Excel, PowerPoint).
<code>O2P_E_PASSWORD = 0x82410C0E</code> Password required	The input document is password protected, or the specified password is incorrect.
<code>O2P_W_DECRYPTERROR = 0x02410C0E</code>	Content decryption error (mail)
<code>O2P_E_OCR = 0x82410C0F</code> OCR error (engine not available)	Verify that the specified OCR engine is available on the server.
<code>O2P_E_PDFACOMPLIANCE = 0x82410C10</code> PDF/A conversion failed to reach required conformance level	Structure information may be missing in the input document, preventing conversion to PDF/A-1a.
<code>O2P_I_VOIDRESULT = 0x02410C11</code> Conversion was successful, result is empty	This code can be returned by a plugin, indicating that it will handle further processing of the output.

## Error codes

O2P_E_PLUGINERROR = 0x82410C12 An unhandled error occurred in a plugin	The log file may contain more information.
O2P_E_VBAERROR = 0x82410C13 VBA error in document	The document contains a macro that caused an error.
O2P_E_APPBLOCKED = 0x82410C14 office application was blocked or failed	The application used to convert a document crashed or was blocked.
O2P_W_SIGN = 0x02410C15 document could not be signed	An error occurred while trying to sign the document.
O2P_E_POPUPBLOCKING = 0x82410C16	Application pop-up dialog box cannot be closed
O2P_W_OPTIMIZE = 0x02410C17	PDF optimization failed
O2P_E_HTTPSTATUS = 0x82410C18	HTTP error while accessing source document.
O2P_W_DOWNGRADE = 0x02410C19 Downgrade during PDF/A conversion	Warning if conformance downgraded during PDF/A conversion.
O2P_W_INPUTSIGNED = 0x02410C1A	Input document is signed (warning)
O2P_E_INPUTSIGNED = 0x82410C1A	Input document is signed (error)
O2P_E_INPUTCORRUPT = 0x82410C1B	Input document corruptions detected
O2P_E_RENDERINGFAILED = 0x82410C1C	Rendering of document failed (e.g. XFA rendering via Acrobat Reader)
O2P_W_NOTPDFA = 0x02410C1D	Result is not PDF/A
O2P_W_UPGRADE = 0x02410C1E	PDF/A conformance was upgraded to PDF/A-2

Other error codes originating from Windows can also be returned. As an example, a code of 2 may be returned from the Windows file system (The system cannot find the file specified).

### Note:

- It is important to note that PDF/A conversion, OCR recognition and digital signing is performed at the end of job processing (i.e. during [JobClose](#)).
- Errors related to OCR and digital signing are often due to configuration issues, while PDF/A conversion problems become manifest due to the specific input.

### Common reasons for PDF/A conversion to fail

- Invalid/corrupted XMP Metadata.
- No suitable font is available to be embedded.
- The document contains transparency features.
- The document contains layers (optional content feature).

- Appearance objects of annotations are missing.
- Unknown or prohibited annotation types.
- Embedded files.

Possible problems may also arise, if the original document contains Javascript resources or digital Signatures. Javascripts are removed from the document, which normally should not have any negative effects.

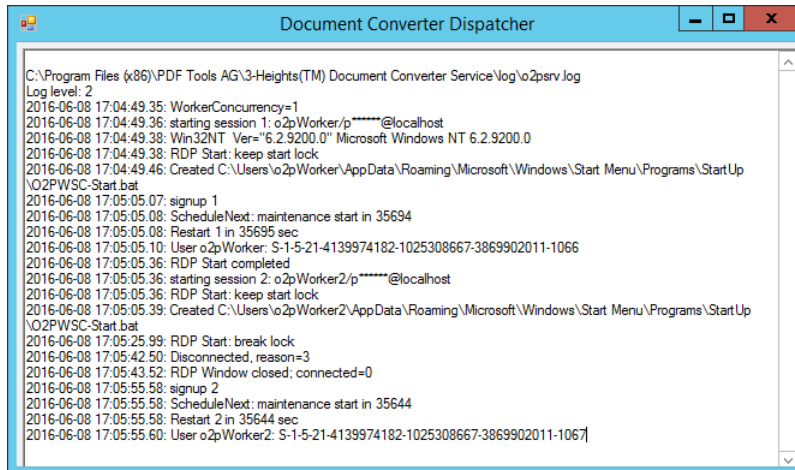
Digital signatures however are inherently invalidated by the PDF/A conversion process. It may thus make sense to remove such signatures prior to passing any PDF files for conversion.

# 7 Debugging

## 7.1 Debugging of the dispatcher service

Starting problems related to the dispatcher service can be debugged in the following way:

1. Make sure the O2PSRV service is stopped (with “Task Manager” or within “Command Prompt” type: `sc stop O2PSRV`).
2. Set the “Worker Count” to 1, see [Service configuration tab](#).
3. Start the service and restart it in debug mode, see again [Service configuration tab](#).



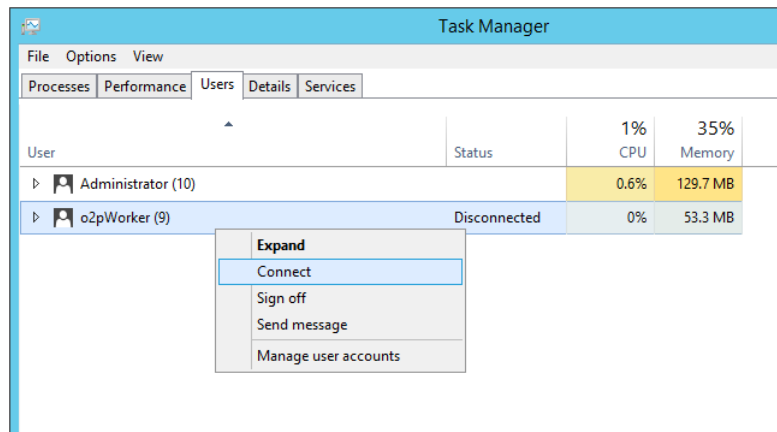
You should see the remote desktop windows pop up, showing an automatic login with the configured Worker Account credentials. After completion of the login, the work session controller should automatically be started.

Common problems that can be detected this way are errors in the configuration of the service account credentials or the remoting configuration of the server. To terminate the converter dispatcher, select the “Document Converter Dispatcher” window and press “Esc”.

### 7.1.1 Debugging of worker sessions

Worker sessions can be debugged similar to the dispatcher service, but make sure all Worker Accounts which are used with the Document Converter are enabled.

**Alternatively** Connect via RDP to the running Worker Session. This can be done with the “Task Manager” on the “Users” tab. Right click on the Worker Account and try to connect to it.



If connected successfully to the Worker Account “disconnect” or connect to the previous session again via the “Task Manager”.

## 7.2 Options

The O2PWSC.ini file supports the following settings for debug use in the [Debug](#) INI-file section.

### 7.2.1 [Debug] INI-File Section

#### DELAYEDCLOSE

**Key:** DELAYEDCLOSE Default: false

Wait two seconds before closing an application popup (default false; no delay).

#### CLOSECRASHBOXES

**Key:** CLOSECRASHBOXES Default: true

Set to false, do not close pop-up messages shown by windows when an application crashes (default true: close boxes).

### 7.2.2 Debug job option

#### KeepTempFiles

**Key:** KeepTempFiles Default: false

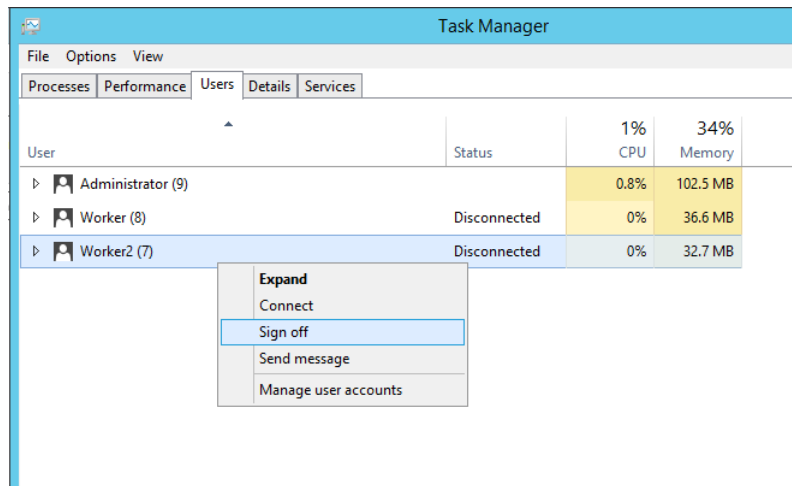
Set this job option to true to suppress deletion of temporary files. This can be useful to inspect files that are temporarily created e.g. as partial conversion results in the folder Temp in the installation directory.

Set this option directly by using the [SetOptions](#) method or in a feature specific way as described in the [User guide](#).

## 8 Troubleshooting

This section lists a number of common problems and points to possible sources to help resolve them.

**Worker related issues** For issues related to the Worker Accounts follow first the steps in section [Debugging](#) to verify the account credentials and remote settings. After that stop the service and make sure that all Worker Accounts are logged off, this can be done with the “Task Manager”. Then restart the service.



### **The Document Converter works for converting an image to PDF, but when I pass a Microsoft Word document, I get the error message “No worker session available”**

Image conversion does not require a worker session, unlike other document types that need an Office application controlled by a worker session.

- Verify that a proper service account for at least one worker session has been configured.
- Verify if any problems occur during starting the worker session; if automatic login fails, verify domain, user and password.

**Conversion of PowerPoint documents fails (no pages converted)** Make sure that background printing is disabled (“Options menu” → “Printing”).

**Our Microsoft Word documents containing the company logo do not get converted correctly** Company logos are usually based on a specific font that needs to be installed on the computer where the conversion takes place. Verify this by inspecting the Fonts directory on the server (usually C:\WINDOWS\Fonts).

**Our application using the Java API crashes** The Java API relies on the .NET transport for accessing the Document Converter. Make sure the O2PProxyNET.dll is registered in the Global Assembly Cache. Use the Windows file explorer to drag and drop the O2PProxyNET.dll from the installation directory to the C:\WINDOWS\assembly folder.

**The Visual Basic Script sample does not work** A common cause for this issue is that O2PProxyNET.dll is not registered in the Global Assembly Cache (see above [Our application using the Java API crashes](#)).

**HTML conversions do not work** Make sure “Internet Explorer Enhanced Security” is configured “Off”. (Start “Server Manager”, under “Security Information”, click “Configure IE ESC”).

**VBAOff setting has no effect** Microsoft Office have the VBAOff flag under \Software\Policies\Microsoft\Office, where the user may not have write access.

Use REGEDIT . EXE, either

- a. Configure write access for the users configured as worker accounts.
- b. Directly set the VBAOff flag in the "Common" key.

**Web service error 404** When converting large documents via the web service, the caller gets an HTTP 404 error.

**Solution** Adjust the maxRequestLength and maxAllowedContentLength settings in the web.config file. Allow sufficient values to account for encoding and SOAP envelope size.

**IEPRINT exit 2** When converting large HTML documents this error may show up. It means that the print operation has timed out. To allow for a longer timeout, you can create an application configuration file IEPRINT.exe.config and set a value for PrintTimeout (30 is the default, which is multiplied with the size of the input file in MB). Please also check the HTMOPT option and configure it as needed.

**Outlook conversion hangs** There are multiple reasons why Outlook may blocks when converting certain MSG documents:

- a. The Outlook profile for the worker user has not yet been initialized: see section [Platform preparations](#).
- b. A "programmatic access" pop-up dialog box may block Outlook: see section [Platform preparations](#).
- c. The MSG document may contain links to external images that slow down conversion: suppress image download by disabling all outgoing connections from Outlook.exe in the firewall.

**Note:** Trust Center settings have no effect, despite Outlook help suggest so.

# 9 Version history

## 9.1 Changes in versions 6.19–6.27

- **Update** license agreement to version 2.9
- **Improved** auto font size feature for text stamping feature.
- **Improved** conversion of emails (TNEF part detection, removal of invalid image references)
- **Fixed** an issue with job object lifetime control
- **New** IEPRINT option value to use Google Chrome browser for HTML conversions
- **Improved** logging from the configuration editor for troubleshooting mail server connection issues
- **Improved** control over print resolution settings via PRINTDM option
- **New** option LoadAllPages for having MSWord pre-load all pages prior to generating output
- **Improved** handling of directory creation before creating temporary files
- **Improved** logging with additional info on unembedding MIME-embedded items during email conversion
- **Improved** stability of htmopt downloads by limiting the number of concurrent downloads and added retry mechanism (configurable via MaxThreadPoolSize and BaseRetryDelayMilliseconds in htmopt.exe.config)

## 9.2 Changes in versions 6.13–6.18

- **Changed** behavior of option PDF.SUBSET. When using the value false, embedded fonts are only replaced if they match the installed font.
- **New** support for image format HEIC/HEIF.
- **New** configuration option to control execution of Javascript in Acrobat Reader
- **Improved** DateFormat option for email conversion (can be used as document or job option, culture selection)
- Stamping
  - **New** attribute `layer` of `<stamp>` to create stamp whose visibility can be controlled by layer.
  - **New** value `transverse` for attribute `align` of `<stamp>`.
  - **New** variable text element `PageNumber`.
  - **New** variable text elements for document metadata properties, e.g. `Author` or `Title`.
  - **New** prefix `^` for page numbers in attribute `page` of `<stamp>` to count from back of document.

## 9.3 Changes in versions 6.1–6.12

- Added Permission Flags list for option PDF.PERMISSION
- **Improved** default value of OCR.TAGGING to detect automatically whether tagging information should be added or not.
- Added file extensions .POT and .POTX to default extensions for PowerPoint
- Digital Signatures
  - Swisscom All-in Signing Service
    - **New** support for accounts (`Identity`) based on Swisscom CA 4 Certificate Authorities.
    - **New** support to create PAdES signatures (format `ETSI.CAdES.detached`).
  - **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 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`.

- Stamping
  - **New** value `shrinkRelToA4` for attribute `flags` of `<stamp>`.
- **Changed** requirement for .NET Framework to 4.7.
- **New** section [OCR](#) added to chapter Processing Guide with use cases and samples.
- **New** support for using modern authentication with Microsoft Exchange Online

## 9.4 Changes in version 5

- **New** additional supported operating system: Windows Server 2019.
- **Changed** behavior when reading a TIFF. The value `Relative` from tag `ResolutionUnit` is now interpreted as `Inch`.
- **New** option to set a border when converting an image to PDF.
- **New** additional supported Office version: Microsoft Office 2019.
- **New** additional OCR processing options `OCR.TEXTMODE`, `OCR.PAGEMODE`, `OCR.IMAGEMODE` (replaces `PDFA.OCRMODE`)
- **New** option `UpdateFieldsAtPrint` to control field update behavior when converting via MS Word's print function
- **Improved** rendering of multiple spaces in HTML mail bodies (as commonly used for indenting)
- Digital Signatures
  - **New** support to get CRLs using HTTPS and via HTTP redirection.

## 9.5 Changes in version 4.12

- **New** HTTP proxy setting in the GUI license manager.
- **Introduced** license feature `Enterprise`.
- 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.
- **Improved** email header template processing with new support for displaying importance and sensitivity information.
- **Improved** HTML optimization with regard to download of external images
- **Improved** HTML optimization to disable IE compatibility mode and permit use of HTML5 features
- **Improved** repair functionality for irregular HTML content in email bodies
- **Improved** logging (daily rotation, output buffering)
- **Improved** conversion and merging of PDF 2.0 documents
- **New** support for factur-x embedding
- **New** support for creating timestamp signatures
- **New** option `FlattenFormFields` to flatten PDF form fields on conversion
- **New** options `AlternateText` and `LanguageTag` for use with image to PDF conversions
- **Improved** worker start procedure (disconnect unrelated RDP sessions to overcome session limit)
- **Improved** documentation (Document Converter related Windows registry keys and policy settings)

## 9.6 Changes in version 4.11

- **New** support for reading and writing PDF 2.0 documents.
- **New** support for the creation of output files larger than 10GB (not PDF/A-1).
- **Improved** repair of corrupt image streams.

- Digital Signatures
  - **New** ability to sign documents that are larger than 2GB (64-bit version only).
- Stamping
  - **New** default compression Flate for PNG images.
- **Improved** MS Word DOCX processing: attached templates are removed before conversion (avoid load attempt when path is not local)
- **New** support for OXPS format
- **New** options TEXT.TITLE and TEXT.LANG for text to PDF conversion (TXT2PDF)
- **Improved** Email header formatting support for finer styling granularity of header items, such as skipped and failed attachments

## 9.7 Changes in version 4.10

- **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.
- **Improved** “direct” conversion of MSG files (without using MS Outlook)
- **Improved** HTML optimization
- **New** 64-bit installer package
- 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 timestamps.
  - **New** signature algorithm RSA with SSA-PSS (PKCS#1v2.1) can be chosen by setting the provider session property `SigAlgo`.
- Stamping
  - **New** attribute `flags` of `<stamp>`, e.g. to create modifiable stamps or stamps that are only visible when printing.
  - **New** attribute `src` of `<image>` allows a HTTP URL or file path.

## 9.8 Changes in version 4.9

- **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.
- **New** Document Converter Client application
- **New** `msg2eml` conversion tool, improves the performance and stability of Outlook Mail conversions.
- **New** **HTML Optimization Tool**, improves the performance and flexibility for HTML conversions.

## 9.9 Changes in version 4.8

- **Changed** option `TRANSFORM`: Add additional parameters as command line arguments to the transform process call.
- **Changed** option `FitToPage` for Excel: Fit the width of the document or fit width and height of the document.
- **New** option `ShowComments`: Show or hide comments from Word documents in the output document.
- **New** feature: Basic support for PDF collections, the first document will be converted.
- **New** feature: Validate Office documents (format 97-2003) using the Microsoft Office Binary File Format Validator.

- **New:** Support Windows Server 2016 (RDP security).
- **Changed** option MAILHEADER: New value `attach`, the mail header will be embedded (PDF output) or added as text document (ZIP output).
- **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.

# 10 Licensing, copyright, and contact

Pdftools (PDFTools AG) is a world leader in PDF software, delivering reliable PDF products to international customers in all market segments.

Pdftools 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® Document Converter is copyrighted. This user manual is also copyright protected; It may be copied and distributed provided that it remains unchanged including the copyright notice.

## Contact

PDF Tools AG  
Brown-Boveri-Strasse 5  
8050 Zürich  
Switzerland  
<https://www.pdf-tools.com>  
[pdfsales@pdf-tools.com](mailto:pdfsales@pdf-tools.com)

# A Background information

## A.1 Configuration background information

This section provides information to help taking decisions when configuring the Converter Service and particularly the number of worker sessions. Here are some of the relevant facts:

**The number of conversions being processed concurrently within a worker session is limited** The Dispatcher Service will not allocate a worker session that already is busy with two conversions.

**The hardware resources available impose certain limits** Office applications remain loaded in memory; they can consume considerable amounts of memory while processing large documents.

**The computer on which a worker session executes can be individually configured** It could be a virtual computer hosted in Microsoft Virtual PC or Virtual Server.

### If your installation needs to process Outlook .MSG files

- Outlook can only run once per user on specific server. This limitation is due to the fact that the Outlook application locks a user specific file (see <http://support.microsoft.com/kb/291636>).
- Outlook needs an exclusive reservation of a PDF Producer printer while outputting a mail message (.MSG file). This is because the "PrintOut" method does not support printing to a file.
- If the .MSG files constitute a small fraction of processed documents, a single worker session will be sufficient. This is the default unless you set the `SingleOutlook` setting to `false` in the `O2PSRV.exe.config` file.
- If .MSG files are frequent, you should configure distinct user accounts for worker sessions running on the same computer. Also make sure that each session will be using its own pool of PDF Producer printer drivers (no issue with a standard installation).
- If either Outlook or Visio documents must be processed, make sure to configure at least two PDF Producer printer instances for each worker session.

**Setting a specific printer in Visio can crash the application** Therefore a specific printer instance must be available that can be set as the default printer for the active user account.

### A.1.1 TIFF output format

When TIFF is selected as the output format, office applications will print to the TIFF Printer driver instead of the PDF Printer driver. Built-in formats like PDF or image formats will be converted directly.

The TIFF format actually offers many options. The choice between color and gray level output is supported by the `PRINTCOLOR` option, but many other parameters are left to be tuned and can be inspected or changed in the Printing Defaults page of the corresponding TIFF Printers.

E.g. the resolution for color TIFF output is controlled by the DPI setting on the T1A and T1B etc. printer entries.

Or the resolution for monochrome TIFF output is controlled by the settings on the T1N and T1O etc. printer entries.

Similarly, the image compression type can be configured via these settings

**Note:**

- If TIFF (and PDF) printer settings are changed, this should be done consistently for all entries.
- Values are propagated from the system defaults to the user printing defaults; changing the system defaults immediately after installation will have the effect that all worker sessions will inherit these values.
- Once a worker session has used a printer entry, it will continue to use the copy of the printer settings of the user account under which the worker session is running. Since system settings are adapted from the Worker by the first time the printer is used. Moreover, the Worker adapts at this point certain settings from `O2PWSC.ini` file.

The Document Converter selects a TIFF Printer based on the color setting only. To change the settings for black and white conversion e.g. from “bi-tonal” to “gray scale” or change the DPI setting, you set the printing defaults of all non-color TIFF Printer entries accordingly. These would be the entries carrying T1N, T10, T2N, etc. in their name.

As built-in conversions do not make use of the TIFF Printers, they use their own options, which can be set at the document or job level. See the options

- |   |                                      |  |
|---|--------------------------------------|--|
| ■ <a href="#">CMRQUAL</a>               | ■ <a href="#">TIFF.COMPR.INDEXED</a> | ■ <a href="#">TIFF.RENDERINGMODE</a>   |
| ■ <a href="#">TIFF.COMPR</a>            | ■ <a href="#">TIFF.COMPR.PDF</a>     | ■ <a href="#">TIFF.ROTATE</a>          |
| ■ <a href="#">TIFF.COMPR.BITON</a>      | ■ <a href="#">TIFF.BPI</a>           | ■ <a href="#">TIFF.UNPACK</a>          |
| ■ <a href="#">TIFF.COMPR.CONTINUOUS</a> | ■ <a href="#">TIFF.DITHERINGMODE</a> | ■ <a href="#">TIFF.XDPI, TIFF.YDPI</a> |

## A.1.2 Conversion of Internet mail messages

MIME conforming mail messages are converted as follows:

1. The HTML body part is extracted, as well as any images that are included in the message and referenced from the HTML page. If no HTML body is present, the text part is extracted.
2. From the mail header, the Subject and From fields are extracted and inserted into the top of the extracted message. All headers are optionally appended to the end of the message.
3. All above parts are passed to the application configured for .HTML documents for rendering to PDF. Since the default of the application option `IEPRINT=true`, the Internet Explorer’s Print method is used.
4. Attachments of the mail message are extracted and also converted to PDF.

**The following options are recognized during mail conversion** [MAILPARTS](#), [TEXTBODY](#), [MAILHEADER](#)

**Encrypted mails can be converted** If appropriate certificates and private keys are installed on the server.

**Signatures of signed mails** Are verified, and the certificate status found during verification is added to the “From” information. The Windows certificate store is used to locate trusted issuer certificates.

All store types are searched that are accessible to the converter process performing the conversion. Therefore, you may want to store relevant certificates either in the “Computer” store, or in each worker user store.

## A.2 Architecture background

### A.2.1 Processing paradigm

The 3-Heights® Document Converter supports processing for single documents and also for converting multiple documents into a single output PDF.

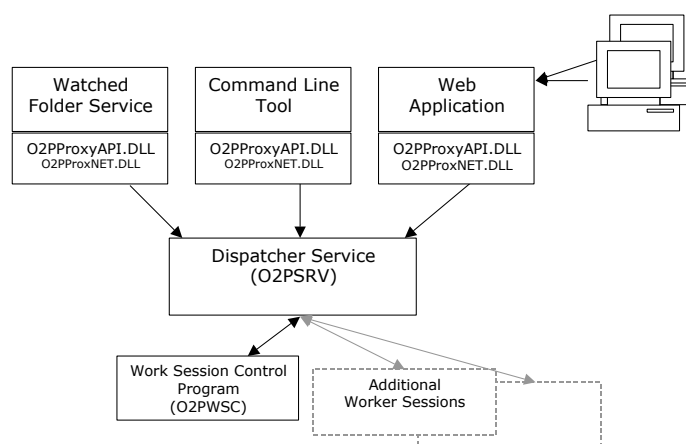
Processing starts with the creation of a job. At this time, the name of the output PDF is specified, and any job specific options are passed. Following that, documents are added one by one to be converted to PDF and appended to the output PDF.

Single document and job processing modes work interleaved, because any single document can actually be an aggregate<sup>3</sup>, containing internally multiple documents of potentially different types, and job processing builds on top of the conversion of individual documents.

When passing documents for processing, they can be accompanied by document related options. For more detailed information see the [3-Heights® Document Converter API documentation](#).

### A.2.2 Architecture

This section presents an overview of the architecture and components of the Document Converter. This information will be helpful for troubleshooting as well as for planning and implementing advanced server configuration involving multiple computers.



**The service executable O2PSRV . exe** Controls startup, job dispatching and shutdown. Typical installations will have this executable registered as a service and configured to start automatically at system start. This process runs detached from a desktop or screen display.

**The session controller executable O2PWSC . exe** Runs in a converter service session to control the office applications in an interactive (-like) environment. One or more such sessions can be configured to accept and process conversion jobs on behalf of the central dispatcher process (O2PSRV). Typically, sessions running O2PWSC will be opened automatically via RDP (Remote Desktop Protocol) when O2PSRV starts up, and terminate again when O2PSRV shuts down.

O2PSRV and O2PWSC will usually run on the same server computer, but could also be distributed to several computers in a network. This can be useful for load balancing or for operational reasons.

<sup>3</sup> For example, emails with attachments or ZIP files.

### The following components will run on the client or middle tier

- The watched folder service executable O2PWFS . exe, as well as the email folder service (if installed)
- The command line executable O2PCLIENT . exe
- The GUI client application O2PCMAApp . exe
- The proxy interface O2PProxyAPI . dll

O2PWFS and O2PCLIENT actually use O2PProxyAPI.DLL to pass document conversion jobs to the Document Converter. While O2PWFS collects polls the drop directory for documents to convert, the command line tool processes documents as specified on the command line. O2PProxyAPI . dll exposes the service interface for these programs, just as it would for any custom application program. This DLL communicates with the Document Converter's dispatcher process (O2PSRV) via the O2PProxyNET . dll.

### Adhoc Converter Service

The "Adhoc Converter Service" consists of a scaled down configuration, where the dispatcher service and the session controller are "merged" into a single process. Execution is initiated by starting O2PWSC . exe with the command line option -x. The execution environment is the current user session from which it is started.

Logging to the event log will not work. No additional worker sessions are started.

The "Adhoc Converter Service" cannot start up successfully when the dispatcher service is running due to conflicting access to the configured network port. Make sure the dispatcher service is stopped (or disabled).

## A.3 Windows policies and registry settings

A number of Windows registry settings are relevant for the operation of the Document Converter. Some of these settings are controlled via Group Policy or Domain Policy settings.

Settings can interfere with the need to automatically start worker sessions via RDP by the document converter (dispatcher) service. The service will try to work around conflicting settings by temporarily changing the settings during RDP connection establishment and restore them afterwards. This will only work if the dispatcher service has sufficient privileges to do so (i.e. an unprivileged account will prevent this mechanism).

### A.3.1 Legal notice caption

The legal notice caption is stored in the registry under HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\legalnoticecaption

The value will be updated from the corresponding policy setting by group policy updates when configured.

Group Policy Editor: Local Computer Policy->Computer Configuration->Windows Settings->Security Settings->Local Policies->Security Options-> Interactive logon: Message text for users attempting to log on

Domain Policy Editor: Computer Configuration->Policies->Windows Settings->Security Settings->Local Policies->Security Options-> Interactive logon: Message text for users attempting to log on

**Note:** If this registry entry is set, the dispatcher service will temporarily unset the registry entry to facilitate the automatic logon when starting a worker session.

### A.3.2 Prompt for password

The Remote Desktop Protocol supports forced password prompting, controlled by the fPromptForPassword flag stored on the server under registry keys HKLM\SYSTEM\CurrentControlSet\Control\Terminal

Server\WinStations\RDP-Tcp and/or HKLM\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services. The latter key reflects the group policy setting "Always prompt for password upon connection" (Group Policy Editor: Computer Configuration->Administrative Templates->Windows Components->Remote Desktop Services->Remote Desktop Session Host->Security).

It is important that this policy is not enabled, as it will block automatic logon of worker sessions started by the dispatcher service.

### A.3.3 Security layer

RDP connections can be required to comply with the requirements of specific security levels as defined by the registry setting HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp\SecurityLayer

The dispatcher service will temporarily lower that level to facilitate the establishing of the local RDP connection when starting a worker session.

### A.3.4 Disabled allow list

The HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Terminal Server\TSAppAllowList\fdisabledAllowList registry setting is relevant when the Remote Desktop Service role is enabled on the server. The server will not allow the execution of a start program upon RDP session start by default. However, the document converter depends on having the worker session control application (O2PWSC.exe) being started via this mechanism. It therefore temporarily modifies this setting during worker startup (only if Remote Desktop Services role is actually installed).

### A.3.5 Remote Desktop Connection settings

A set of settings related to RDP connections exists that can potentially interfere with the requirements of the document converter needing stable execution and control over its worker sessions. The individual settings can be stored on the local server, in group or domain policies, or in the default user configuration. The corresponding registry locations are:

- SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services
- SYSTEM\CurrentControlSet\Control\Terminal Server\DefaultUserConfiguration
- SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp

#### MaxDisconnectionTime

This setting limits the maximum duration of an RDP session being disconnected. To avoid worker sessions to be killed inadvertently, the options are either to allow an unlimited duration, or to keep worker sessions connected.

Keeping worker sessions connected in turn requires that the Remote Desktop Service role be enabled and a sufficient count of RDP CAL licenses be procured and installed.

The Document Converter will write a warning to the log file if the setting enabled.

#### MaxConnectionTime

This setting limits the maximum duration a session can be connected via RDP. Make sure there is no such limit.

The Document Converter will write a warning to the log file if the setting enabled.

## MaxIdleTime

This setting limits the maximum time a session is left idle by the user. Make sure to not enable this setting. The Document Converter will write a warning to the log file if the setting enabled.

## RemoteAppLogoffTimeLimit

This setting affects session termination by Windows when a remote application process ends. The setting is monitored by the document converter to write a warning to the log file if enabled.

## fResetBroken

This setting controls what happens with timed out sessions. See description in policy editor for "End session when time limits are reached" (Remote Desktop Services->Remote Desktop Session Host->Session Time Limits).

The document converter monitors this setting and writes a warning to the log file if enabled.

## A.4 Installation parameters

The installation can be controlled by specifying values for the public properties of the installer package (see Microsoft documentation for `msiexec.exe`)

The server host name and the port number are properties that can also be set on the Windows Installer (`msiexec.exe`) command line. The names of these properties are `SERVERNAME` and `SERVERPORT`, respectively.

**Example:** Specify custom values for the public properties `SERVERNAME` and `INSTALLLEVEL`

```
MSIEXEC /quiet /i "3-Heights(TM) Document Converter Service.msi"  
SERVERNAME=dcs.mycompany.com INSTALLLEVEL=100
```

### Useful public properties

Property	Description
INSTALLLEVEL	Numeric value controlling the selection of features to be installed; <b>Default:</b> 100. The feature levels of the individual features are: <b>Server Installation Core Components</b> 1 <b>Client App</b> 20 <b>Command Line Tool</b> 40 <b>Watched Folder Service</b> 200 <b>Mail Folder Service</b> 250 <b>Software Development Kit</b> 300 <b>Web Service</b> 350
INSTALLDIR	The installation directory

## Useful public properties

SERVERNAME	The network name of the computer hosting the Document Converter Service (Dispatcher); <b>Default:</b> localhost
SERVERPORT	The port number on which the Document Converter Dispatcher listens; <b>Default:</b> 7981
ADDLOCAL	Install features by setting the name (always set ServerInstallation!) <ul style="list-style-type: none"><li>■ ServerInstallation</li><li>■ ClientApp</li><li>■ CommandLineTool</li><li>■ WatchedFolderService</li><li>■ MailFolderService</li><li>■ SDK (Software Development Kit)</li><li>■ WebService</li></ul>

Please refer to the Microsoft Windows Installer Guide for detailed information about other standard public properties.

## A.5 Stamp file

### A.5.1 Stamp file syntax

Stamps are described with XML data that is passed to the 3-Heights® Document Converter as file using the option `-s`. A stamp file can contain one or more stamps.

For each Tag there is a separate table below, where the **Attribute-Names** and the **Attribute-Values** are described.

#### <pdfstamp>

The **Root Tag** for the PDF stamp XML file. The tag may contain multiple stamps.

**xmlns="http://www.pdf-tools.com/pdfstamp/"** (required)

XML namespace used for all stamp elements.

### Stamp

A stamp is defined by a `<stamp>` tag that specifies the stamp's size, position, and pages to which it is applied to. The stamp's appearance is defined by the content operators contained therein.

#### <stamp> Add a Stamp

**page="<page\_set>"** (required)

The pages to which the stamp is to be applied. The syntax is as follows:

`<page_set> = <page_range> [", " <page_range>]`

`<page_range> = <n> | <n1>-<n2> | first | last | not_first | not_last | even | odd | all`

Where:

- `<n>`, `<n1>`, `<n2>`: Page number. **1** defines the first page.

The prefix `^` can be used to count from the end of the document. For example, `^1` specifies the last and `^2` the second to last page.

- `first`: First page
- `last`: Last page
- `odd`: Only odd pages including first page and last page in case it is odd
- `even`: Only even pages including last page in case it is even
- `all`: All pages
- `not_first`: First page excluded
- `not_last`: Last page excluded

Example: `page="1,2-4,6,10,last"`

`name="<identifier>"` (optional)

Unique identifier of the stamp, must be less than 127 characters, see section [Modify content of existing stamps](#) for more information.

`relativepos="<x> <y>"` (required)

Relative position `<x>` and `<y>` of the stamp with regards to the page. Positive values of `<x>` and `<y>` define the distances of the stamp to the left and lower, negative values to the right and upper page boundary respectively. The units of the values are PDF units of 1/72 inch. The positioning algorithm works best for stamp rotation angles that are a multiple of 90° (see `rotate` attribute).

`<x>` or `<y>` are ignored, if respective `align` is used.

Examples:

1. `relativepos=" 10 -10"` places the stamp in the upper left corner of the page.
2. `relativepos="-10 -10"` places the stamp in the upper right corner of the page.
3. `relativepos=" 10 10"` places the stamp in the lower left corner of the page.
4. `relativepos="-10 10"` places the stamp in the lower right corner of the page.

`align="<alignment>"` (optional)

Align the stamp with the page. Allowed values for `<alignment>` are:

- `center`: position horizontally at center of page, the `<x>` value of `relativepos` is ignored.
- `middle`: position vertically at middle of page, the `<y>` value of `relativepos` is ignored.
- `transverse`: position stamp in the middle of the page and rotate it, such that it aligns with the diagonal of the page from the lower left to the upper right corner. Note that `transverse` cannot be used in combination with the attributes `relativepos` and `rotate`.

Examples:

1. `<stamp position="0 4" align="center">`  
Centers the stamp horizontally and 4 pt away from the bottom of the page.
2. `<stamp position="-4 0" align="middle">`  
Centers the stamp vertically and 4 pt away from the right edge of the page.

`size="<w> <h>"` (optional)

The width and height of the stamp. The stamp's content will be clipped to this rectangle. If this is not specified or either `<w>` or `<h>` are zero, the respective size is calculated to fit content.

`rotate="<angle>"` (optional)

Rotation of the stamp in degrees clockwise.

`scale="<scale_set>"` (optional)

Modify scale of stamp. Allowed values for `<scale_set>` are:

- `relToA4`: Scale the stamp relative to the page size. For example, make stamp half as large on an A5 and twice as large on an A3 page as specified.

- **shrinkRelToA4**: Shrink stamp for all pages smaller than A4. For example, on A5 make stamp half as large as specified and as specified an A3 page.

**autoorientation="<b>" (optional)**

Allowed values for <b> are:

- **false (default)**: Always position stamps as defined by stamp attributes.
- **true**: Detect orientation (portrait and landscape) of page automatically and treat landscape page as 90° rotated portrait. Useful to apply stamps to "long" or "short" edge of page.

**alpha="<ca>" (optional)**

The opacity of the stamp as a whole. **1.0** for fully opaque, **0.0** for fully transparent.

Default: **1.0**

The PDF/A-1 standard does not allow transparency. Therefore, for PDF/A-1 conforming input files you must not set alpha to a value other than **1.0**.

**type="<type>" (optional)**

The type of the stamp

- **annotation (default)**: The stamp is added to the page as a stamp annotation. Creating or modifying stamps of this type will not invalidate existing signatures of the input document. While it is not easily possible to remove stamps of this type, it is possible to print a document without annotations.
- **foreground<sup>4</sup>**: The stamp is added to the foreground of the page content. Creating or modifying stamps of this type will invalidate all existing signatures of the input document. It is not easily possible to remove stamps of this type nor can the document be printed without them.
- **background**: The stamp is added to the background of the page content. Creating or modifying stamps of this type will invalidate all existing signatures of the input document. It is not easily possible to remove stamps of this type nor can the document be printed without them.

Note that stamps placed this way can be hidden when pages contain a non-transparent background. In these cases, you may rather want to put the stamps in the foreground, but apply alpha transparency to achieve a result with existing content not covered completely.

**flags="<flags>" (optional)**

Set the flags of the stamp annotation (i.e. stamps with **type="annotation"**). <flags> is a comma separated list of the following values: **NoView**, **Print**, **ReadOnly**, and **Locked**. See chapter 8.4.2 "Annotation Flags" of the [PDF Reference 1.7](#) for a description of the flags.

For PDF/A conformance, the flag **Print** must be set and **NoView** must not be set.

Default: **Print**, **ReadOnly**, **Locked**

**layer="<name>" (optional)**

Set the name of the layer that can be used by the consumer to selectively view or hide the stamp. If the attribute is omitted or its value is empty, no layer is used so the stamp is always visible.

For input documents that already contain a layer of the specified name the document's existing layer is used. Otherwise, a new layer is created. The new layer is visible by default and inserted at the end of the document's list of layers.

Default: no layer

The PDF/A-1 standard does not allow layers. Therefore, for PDF/A-1 conforming input files you must not set the attribute **layer**.

<sup>4</sup> Up to version 4.5.6.0 of the 3-Heights® Document Converter this type was called **content**.

## Coordinates

All coordinate and size values are in PDF units of 1/72 inch (A4 = 595 x 842 points, letter = 612 x 792 points). The origin of the coordinate system is generally the lower left corner of the reference object. For stamps the reference object is the page, for content operators the reference is the stamp rectangle.

## Modify content of existing stamps

Setting the `name` attribute of a stamp allows the stamp's content to be replaced later. If an existing stamp with the same name exists in the input file, its content is replaced as shown in example [Example 2: Modify "Simple Stamp"](#). Note that when updating a stamp, its pageset, position and size cannot be changed. Therefore, if you intend to update a stamp, make sure to create it specifying a `size` that is sufficiently large.

When modifying a stamp, only its content may be changed. All attributes of `<stamp>` must remain unchanged, in particular `page`, `size` and `type`.

## Stamp content

Each stamp contains a number of content operators that define the appearance (i.e. the content) of the stamp. The content operators are applied in the order they appear within `<stamp>` where each content element is drawn over all previous elements (i.e. increasing z-order).

## Text

Stamp text is defined by `<text>`. All character data (text) therein is stamped:

```
<text font="Arial" size="12">Some text</text>
```

Text fragments can be formatted differently by enclosing them in a `<span>` element. All text formatting attributes are inherited from the parent element and can be overridden in `<span>`:

```
<text font="Arial" size="12" >Text with a <span  
font="Arial,Bold">bold</span> and a <span  
color="1 0 0 ">red</span> word.</text>
```

Note that all character data in `<text>` is added, including whitespace such as spaces and line breaks.

### `<text>` Add Text

All text formatting attributes described in [<span>](#) can also be specified in `<text>`.

`position="<x> <y>"` (optional)

The position in points within the stamp, e.g. "200 300".

With the default values for `align` (`align="left top"`), `position` defines the top left corner of the text<sup>5</sup>.

`align="<xalign> <yalign>"` (optional)

Align text at `position` or stamp, if `position` is not set.

<sup>5</sup> Prior to version 4.4.31.0 of the 3-Heights® Document Converter, `position` specified the origin of the first character. When upgrading, add `0.75*size` to the value of `<y>`.

Values for horizontal alignment `<xalign>`:

- `left`: align to the left (**default**)
- `center`: center text
- `right`: align to the right

Values for vertical alignment `<yalign>`:

- `top`: align to the top (**default**)
- `middle`: align to the middle
- `bottom`: align to the bottom

Examples:

1. `<text align="left bottom" ...>`  
positions the text in the left bottom corner of the stamp.
2. `<text align="left bottom" position="10 10" ...>`:  
align left bottom corner of text to position "10 10".

`format="<b>"` (**optional**)

Whether or not to enable formatting of variable text. Allowed values for `<b>` are `true` and `false` (**default**). See [Variable text](#) for more information.

`text="<text>"` (**optional**)

The text that is to be written, e.g. `text="Hello World"`.

Multi-line text is supported by using the newline character `&#10;`, e.g. `text="1st line&#10;2nd line"`.

If the attribute `text` is not specified, the text content of `<text>` is used. So `<text ... text="Hello World"/>` produces the same result as `<text ...>Hello World</text>`.

### `<span>` Define Formatting of Text

Example: `<text font="Arial" size="8"><span font="Arial,Bold">Note:</span> Text can be formatted using &lt;span&gt;.</text>`

`color="<r> <g> <b>"` (**optional**)

The color as RGB value, where all values must be in the range from 0 to 1, e.g.:

- Red: "1 0 0"
- Green: "0 1 0"
- Yellow: "1 1 0"
- Black: "0 0 0" (**default**)
- Gray: "0.5 0.5 0.5"

`font="<name>"` (**required**)

The TrueType name of the font, e.g. "Arial" or "Times New Roman,Bold", or a complete path to the font, e.g. "C:\Windows\Fonts\Arial.ttf".

TrueType names consist of a font family name, which is optionally followed by a comma and style, e.g. "Verdana,Italic". Commonly available styles are "Bold", "Italic", and "BoldItalic". The respective font must be available in any of the font directories.

`size="<n>"` (**required**)

The font size in points, e.g. 12.

If set to `0` or `auto`, the size is chosen such that text fits the stamp's size. This is only allowed under these conditions:

- The `<text>` element is not within a transformation operator.
- The stamp has a fixed size. It can either be defined by the attribute `size` or from updating an existing stamp.
- If the text's attribute `position` is set, the position must be inside the stamp's size.

#### `fontencoding="<encoding>"` (optional)

This attribute is relevant only, if the stamp will be modified later (see section [Modify content of existing stamps](#)).

The PDF/A standard demands that all used fonts must be embedded in the PDF. Since fonts with many glyphs can be very large in size (>20MB), unused glyphs are removed prior to embedding. This process is called subsetting. The value `<encoding>` controls the subsetting and must be one of the following:

- **Unicode: (default)** Only the glyphs used by the stamp are embedded. If the stamp is modified, a new font that includes the new glyph set has to be re-embedded. This setting is recommended for stamps that will not be modified later.
- **WinAnsi:** All glyphs required for WinAnsiEncoding are embedded. Hence the text's characters are limited to this character set. If the content of the stamp is updated, fonts using `WinAnsi` will be reused.

For example, embedding the font Arial with `Unicode` and approximately ten glyphs uses 20KB while Arial with `WinAnsi` (approximately 200 glyphs) uses 53KB of font data.

#### `mode="<modes>"` (optional)

The attribute `mode` controls the rendering mode of the text.

Allowed values for `<modes>` are the following or a combination thereof:

- **fill: (default)** The text is filled.
- **stroke:** The text's outlines are stroked. The width of the stroke is specified by `linewidth`.

#### `linewidth="<f>"` (optional)

Set the line width in points, e.g. `1.0` (default).

#### `decoration="<decorations>"` (optional)

The attribute `decoration` can be used to add any of the following text decorations:

- **underline:** A small line is drawn below the text.

#### `<link>` Create Link

For all text contained within this element, a link is created. Links work best for stamps with `type="foreground"`, but are possible for other types as well.

Example: `<text font="Arial" size="8">@ <link uri="https://www.pdf-tools.com/">  
> Pdftools - PDF Tools AG</link></text>`

#### `uri="<uri>"` (required)

The URI which is the link target.

#### `<filltext>` Obsolete tag.

Starting with version 4.9.1.0 of the 3-Heights® Document Converter the element `<filltext ...>` was rendered obsolete by `<text ...>`.

#### `<stroketext>` Obsolete tag.

Starting with version 4.9.1.0 of the 3-Heights® Document Converter the element `<stroketext ...>` was rendered obsolete by `<text mode="stroke" ...>`.

## Variable text

Variable text such as the current date or the number of pages can be stamped in `<text>`. The feature must be activated by setting `format="true"`.

Variable text elements are of the following form:

```
"{<value>:<format>}"
```

The `<value>` defines the type of value. `<format>` is optional and specifies how the value should be formatted. To stamp the `{` character, it must be escaped by duplicating it: `{{`.

## String values

`<value>` The following values are supported:

- **Title**: the document's title
- **Author**: the name of the person who created the document
- **Subject**: the subject of the document
- **Creator**: the original application that created the document
- **Producer**: the application that created the PDF

**Example:** Stamp the document author.

Text	Result
Author: {Author}	Author: Peter Pan

## Date values

`<value>` The following values are supported:

- **UTC**: the current time in UTC
- **LocalTime**: the current local time
- **CreationDate**: the date and time the document was originally created
- **ModDate**: the date and time the document was most recently modified

`<format>` The default format is a locale-dependent date and time representation. Alternatively a format string as accepted by `strftime()` can be specified.

**Example:** Stamp the current local time with the default format.

Text	Result
Received: {LocalTime}	Received: Thu Aug 23 14:55:02 2001

**Example:** Stamp the current date.

Text	Result
Date: {LocalTime:%d. %m. %Y}	Date: 23. 8. 2011

## Number values

**<value>** The following values are supported:

- **PageNumber**: the page number. Note that when updating the content of an existing stamp as described in [Modify content of existing stamps](#), the new content can only contain **PageNumber** if the existing stamp also used **PageNumber**.
- **PageCount**: the number of pages in the document

Optionally, an offset can be appended to the **<value>**, where positive offsets start with **+** and negative with **-**. For example **{PageCount+2}** to add or **{PageCount-2}** to subtract 2 from the actual page count.

**<format>** Optionally a format string as accepted by `printf()` can be specified.

**Example:** Stamp the page count.

Text	Result
<code>{{PageCount}} = {PageCount}</code>	<code>{PageCount} = 10</code>

**Example:** Stamp the current date and time onto each page's lower right corner.

```
<?xml version="1.0" encoding="utf-8"?>
<pdfstamp xmlns="http://www.pdf-tools.com/pdfstamp/">
  <stamp page="all" relativepos="-10 10">
    <text font="Arial" size="10" format="true">Date: {LocalTime}</text>
  </stamp>
</pdfstamp>
```

## Images and geometric shapes

### **<image>** Add Image

**rect="<x> <y> <w> <h>"** (required)

The rectangle where the image is to be placed at. **<x>** and **<y>** correspond to the location (lower left corner), and **<w>** and **<h>** to width and height of the image, e.g. `100 200 50 50`

**src="<url>"** (required)

The URL or path to the file<sup>6</sup>, e.g. `C:/pictures/image1.jpg` or `http://www.mydomain.com/image1.jpg`.

**compression="<value>"** (optional)

By default, bitonal images are compressed with **CCITTFax**, continuous tone images with **DCT** and indexed images with **Flate**. To explicitly set the compression, use this property.

Supported values are:

- **Flate**: Flate encoded
- **DCT**: DCT (JPEG) encoded
- **CCITTFax**: CCITT G4 encoded

### **<fillrectangle>** Add Filled Rectangle

<sup>6</sup> Prior to version 4.10.13.0 of the 3-Heights® Document Converter, this attribute was called `filename`.

**rect**="<x> <y> <w> <h>" (optional)

The coordinates and size of the rectangle. If this value is omitted, the rectangle fills the entire area of the stamp.

**color**="<r> <g> <b>" (optional)

The fill color of the rectangle. The color as RGB value, where all values must be in the range from 0.0 to 1.0. The default is black: "0 0 0"

**alpha**="<ca>" (optional)

The opacity of the rectangle. 1.0 for fully opaque, 0.0 for fully transparent.

Default: 1.0

The PDF/A-1 standard does not allow transparency. Therefore, for PDF/A-1 conforming input files you must not set alpha to a value other than 1.0.

### <strokerectangle> Add Stroked Rectangle

**linewidth**="<f>" (optional)

Set the line width in points, e.g. 1.0 (default).

For the following parameter descriptions see [<fillrectangle>](#).

**rect**="<x> <y> <w> <h>"

**color**="<r> <g> <b>"

**alpha**="<ca>"

## Transformations

The transform operators apply to stamp content defined within the tag. For example, this can be used to rotate <text> or <image>.

### <rotate> Rotation

**angle**="<n>" (required)

Rotate by <n> degrees counter-clockwise, e.g. 90

**origin**="<x> <y>" (required)

Set the origin of the rotation in points, e.g. 100 100

### <translate> Coordinate Translation

**offset**="<x> <y>" (required)

The <x> (horizontal) and <y> (vertical) offset in points. A translation by x y is equal to a transformation by  $\begin{matrix} 1 & 0 & 0 & 1 & x & y \end{matrix}$ .

### <transform> Coordinate Transformation

**matrix**="<a> <b> <c> <d> <x> <y>" (required)

The transformation matrix to scale, rotate, skew, or translate.

Examples:

1. Identity: 1 0 0 1 0 0
2. Scale by factor 2 (double size): 2 0 0 2 0 0
3. Translate 50 points to left, 200 up: 1 0 0 1 50 200
4. Rotate by x:  $\begin{matrix} \cos(x) & \sin(x) & -\sin(x) & \cos(x) & 0 & 0 \end{matrix}$   
For 90° (=  $\pi/2$ ) that is: 0 1 -1 0 0 0

## A.5.2 Examples

### Example 1: Simple stamps

Apply two simple stamps.

**First stamp:** Stamp text "Simple Stamp" on in upper left corner of all pages.

**Second stamp:** Stamp image `lena.tif` rotated by 90° and located at the center of the top corner of the first page.

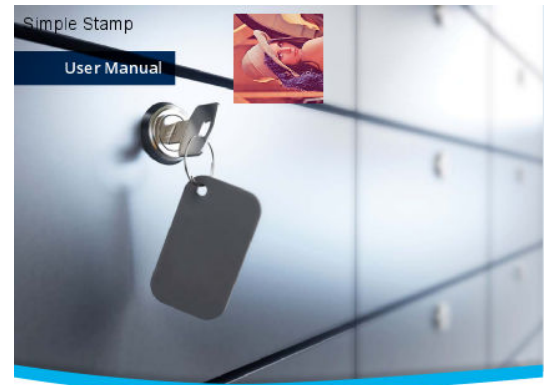
example1.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<pdfstamp xmlns="http://www.pdf-tools.com/pdfstamp">

  <stamp page="all" name="simple stamp"
    relativepos="10 -10" size="160 0">
    <text align="left middle"
      font="Arial" size="20" fontencoding="WinAnsi"
      text="Simple Stamp" />
  </stamp>

  <stamp page="first"
    relativepos="0 -10" align="center">
    <rotate angle="90" origin="50 50">
      <image rect="0 0 100 100"
        filename="C:\images\lena.tif"/>
    </rotate>
  </stamp>

</pdfstamp>
```



3-Heights™  
PDF Security API

Version 4.8.24.0

PDF-TOOLS.COM  
Premium PDF Technology



Result of [example1.xml](#).

### Example 2: Modify "Simple Stamp"

Modify "simple stamp" from [Example 1: Simple stamps](#).

The stamp "simple stamp" can be modified by applying the following stamp XML file to the output file of the example above. Note that since position and size of the stamp remain unchanged, the respective attributes can be omitted.

The second stamp applied in [Example 1](#) is not modified.

### example2.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<pdfstamp xmlns="http://www.pdf-tools.com/pdfstamp/">
  <stamp name="simple stamp">
    <text align="left middle"
      color="1 0 0"
      font="Arial" size="20" fontencoding="WinAnsi"
      text="Modified Stamp" />
  </stamp>
</pdfstamp>
```



## 3-Heights™ PDF Security API

Version 4.8.24.0



Result of [example2.xml](#).

### Example 3: Add watermark text diagonally across pages

The stamp is specified for an A4 page. On each page the stamp is applied to, it is scaled (`scale="relToA4"`) and rotated (`align="transverse"`) to fit the page.

### example3.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<pdfstamp xmlns="http://www.pdf-tools.com/pdfstamp/">
  <stamp page="all"
    scale="relToA4"
    align="transverse"
    type="foreground">
    <text mode="stroke"
      font="Arial,Bold" size="60"
      >WATERMARK TEXT</text>
  </stamp>
</pdfstamp>
```



## 3-Heights™ PDF Security API

Version 4.8.24.0



Result of [example3.xml](#).

## Example 4: Apply stamp to long edge of all pages

Stamp has a light gray background and a black border.

example4.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<pdfstamp xmlns="http://www.pdf-tools.com/pdfstamp/">
  <stamp page="all" size="802 28"
    relativepos="5 0" align="middle" rotate="90"
    scale="relToA4" autoorientation="true"
    alpha="0.75" type="foreground">
    <fillrectangle color="0.8 0.8 0.8"/>
    <strokerectangle/>
    <text align="center middle"
      font="Arial" size="20"
      text="stamp on long edge"/>
  </stamp>
</pdfstamp>
```



Result of [example4.xml](#).

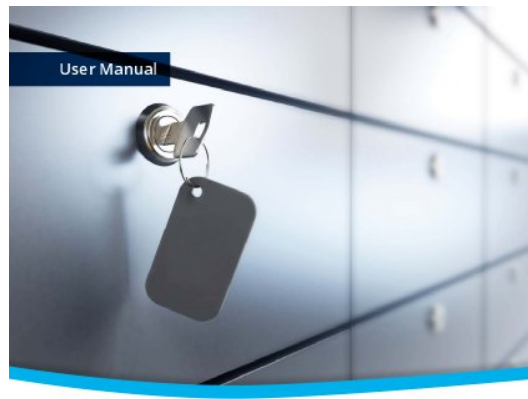
## Example 5: Stamp links

Stamp a list of links.

example5.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<pdfstamp xmlns="http://www.pdf-tools.com/pdfstamp/">
  <stamp page="first" type="content" relativepos="-10 10" >
    <text font="MyriadPro" size="20" >Bookmarks:
  - <span color="0 0 1" decoration="underline"><link
    uri="http://www.pdf-tools.com/...">Product website</link></span>
  - <span color="0 0 1" decoration="underline"><link
    uri="http://www.pdf-tools.com/.../seca.pdf">Manual</link></span>
  - <span color="0 0 1" decoration="underline"><link
    uri="https://www.pdf-online.com/osa/secure.aspx">Online sample</link></span>
    </text>
  </stamp>
</pdfstamp>
```

## Result of [example5.xml](#).



### 3-Heights™ PDF Security API

Version 4.8.24.0



Bookmarks:  
- [Product website](#)  
- [Manual](#)  
- [Online sample](#)

## A.6 Samples

### A.6.1 C

This sample shows the basic call sequence and usage of the API functions. Error handling should follow each call.

```
#include "o2pproxyapi_c.h"
#include "o2perror.h"
void CheckLastError(O2PJob job)
{
    unsigned int iError;
    const char* pszText;
    O2PJobGetLastErrorA(job, &iError, &pszErrorText);
    if (iError)
        fprintf(stderr, "Error code: 0x%x\nError text: %s\n", iError, pszText);
}

int main(int argc, char* argv[])
{
    T02PConverter* converter = O2PCreateConverter(NULL);
    T02PJob* job = O2PConverterCreateJob(converter);
    O2PJobSetOptionsA(job, "#;");
    O2PJobCreateOutput(job, argv[argc-1]);
    if(!O2PJobAppendDoc(job, argv[1], NULL))
    {
        CheckLastError(job);
    }
    O2PJobClose(job);
    CheckLastError(job);
}
```

```
O2PJobDestroyObject(job);
O2PDestroyConverter(converter);
return 0;
}
```

## A.6.2 C# .NET

```
using PdfTools.Converter;

try {
    job = ConverterFactory.GetInstance(
        "tcp://servername:7981/O2PService").CreateJob();
}
catch(System.Net.Sockets.SocketException se) {
    ErrText.Value = "Conversion service not available; " + se.Message;
    return;
}
ErrorInfo ei;
if (!job.SetOptions("PDFA"))
{
    ei = job.GetLastError();
    CheckError(ei); // application specific error handling
}
job.CreateOutput(null);
byte[] DocBytes = System.IO.File.ReadAllBytes("c:\\test.docx");
string DocOptions = "ORIGINALNAME=test.docx";
if (!job.AppendDoc(DocBytes, DocOptions)) {
    ei = job.GetLastError();
    CheckError(ei);
}
job.FinishConversion();
ei = job.GetLastError();
CheckError(ei);
byte[] outBytes = job.RetrieveOutput();
job.Terminate();
```

The above code sample shows an excerpt from an ASP.NET application that uses the .NET interface of the Document Converter.

**Note:** This is the bare remoting interface without any client side logic as in the C or COM interfaces. Retrieving error information e.g. is a remote call here, while the C and COM interfaces are caching that information.

To actually run this code from within a .NET executable or ASP.NET, the O2PProxyNET .DLL must either be explicitly referenced from the application, or it must be registered in the .NET Global Assembly Cache.

## A.6.3 Visual Basic Script

```
Set oConv = CreateObject("O2PProxyAPI.ConverterProxy")
oConv.ServicePoint = "tcp://servername:7981/O2PService"
```

```

Set oJob = oConv.CreateJob()
If oJob.Create("output.pdf") Then
    oJob.AppendDoc "Document1.rtf", ""
    oJob.Close
Else
    WScript.Echo "Create failed: " & oJob.ErrorText
End If

```

**Note:** In order to execute this script, the O2PProxyNET .DLL must be registered in the .NET Global Assembly Cache.

## A.6.4 Java

### Microsoft.NET based Java API

This program sample illustrates the use of the Microsoft.NET based Java interface that is available only on Windows platforms. Java applications that are hosted on other platforms will need to make use of the web service interface.

```

import com.pdfutils.converter.*;
public class DocConv
{
    public static void main(String[] args) throws Exception
    {
        if(args.length < 2) {
            System.out.println("Usage: DocConv in1.pdf in2.pdf... out.pdf");
            return;
        }
        Converter conv = Converter.createConverter("tcp://srv:7981/O2PService");
        Job job = conv.createJob();
        job.createOutput(args[args.length-1]);
        for(int i = 0; i < args.length-1; i++)
            job.appendDoc(args[i], "");
        job.close();
        job.destroyObject();
        conv.destroyObject();
    }
}

```

**Note:** In order to execute this java program, O2PProxyNET .DLL must be registered in the .NET Global Assembly Cache (unless a copy of O2PProcxNET .DLL is located in the same folder as java .exe).

### Web service interface

The “wsimport” tool can be used to generate the Java code necessary to access the web service (see comment in the sample code below).

```

import java.io.FileInputStream;
import java.io.FileOutputStream;

```

```

import java.net.URL;
import javax.xml.namespace.QName;
import com.pdf_tools.ws.o2p.*;
/**
 * This java sample shows how to use the Document Converter web service
 * interface. The com.pdf_tools.ws.o2p package was created using the
 * wsimport tool:
 * wsimport.bat -Xendorsed -extension
 *               -keep http://wshostname/Converter.asmx?wsdl
 */
public class convert {
    public static void main(String[] args) {
        if (args.length < 2) {
            System.out.println
                ("Usage: convert [Options] path-to-input-document path-to-output-file");
            System.out.println("Options:  -j ConverterOptions");
            System.out.println("        -sp
http://<servername>/Converter.asmx?wsdl");
            return;
        }
        String input_file = null;
        String output_file = null;
        String options = "";
        String sp = "https://docws.pdf-tools.com/Converter.asmx?wsdl";
        int nr = 0;
        while (nr < args.length) {
            if ("-j".equals(args[nr]) && nr+1 < args.length)
                options = args[++nr];
            else if ("-sp".equals(args[nr]) && nr+1 < args.length)
                sp = args[++nr];
            else if (input_file == null)
                input_file = args[nr];
            else if (output_file == null)
                output_file = args[nr];
            else
                System.out.println("too many parameters; skipped '" + args[nr] + "'");
            nr++;
        }
        try {
            URL url = new URL(sp);
            Converter conv = new Converter(url,
                new QName("http://pdf-tools.com/ws/o2p/", "Converter"));
            ConverterSoap ws = conv.getConverterSoap12();
            FileInputStream is = new FileInputStream(input_file);
            byte[] inputDocBytes = new byte[is.available()];
            int n = is.read(inputDocBytes);
            if (n != inputDocBytes.length)
                throw new Exception("Error reading bytes for file " + input_file);
            is.close();
            int pos = input_file.lastIndexOf('/');
            if (pos < 0)
                pos = input_file.lastIndexOf('\\');
            String name = input_file;
            if (pos > 0)
                name = input_file.substring(pos + 1);
            options += ";ORIGINALNAME=\"" + name + "\"";
            ConversionResult result = ws.convertFile(inputDocBytes, options, null);
            if(result.getErrorCode() != 0) {

```

```

        System.out.println("Error: " + result.getErrorCode());
        System.out.println(result.getErrorDescription());
    }
    byte[] outputPdfBytes = result.getDocumentBytes();
    if (outputPdfBytes != null) {
        FileOutputStream os = new FileOutputStream(output_file);
        os.write(outputPdfBytes);
        os.close();
        System.out.println("Pages converted: " + result.getNumPages());
    }
} catch (Exception e1) {
    System.out.println(e1);
}
}
}
}

```

## A.6.5 Web service

### WSDL definition

The web service interface features methods to convert a document (or ZIP package of documents) in a single call. Here is the WSDL definition:

```

<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema"
    xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    xmlns:tns="http://pdf-tools.com/ws/o2p/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    targetNamespace="http://pdf-tools.com/ws/o2p/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
<wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace=
        "http://pdf-tools.com/ws/o2p/">
        <s:element name="ConvertFile">
            <s:complexType>
                <s:sequence>
                    <s:element minOccurs="0" maxOccurs="1" name="DocBytes"
                        type="s:base64Binary" />
                    <s:element minOccurs="0" maxOccurs="1" name="Options"
                        type="s:string" />
                    <s:element minOccurs="0" maxOccurs="1" name="Metadata"
                        type="s:base64Binary" />
                </s:sequence>
            </s:complexType>
        </s:element>
        <s:element name="ConvertFileResponse">
            <s:complexType>
                <s:sequence>
                    <s:element minOccurs="1" maxOccurs="1" name="ConvertFileResult"
                        type="tns:ConversionResult" />
                </s:sequence>
            </s:complexType>
        </s:element>
    </s:schema>
</wsdl:types>

```

```

    </s:sequence>
  </s:complexType>
</s:element>
<s:complexType name="ConversionResult">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="DocumentBytes"
      type="s:base64Binary" />
    <s:element minOccurs="1" maxOccurs="1" name="NumPages"
      type="s:int" />
    <s:element minOccurs="1" maxOccurs="1" name="ErrorCode"
      type="s:int" />
    <s:element minOccurs="0" maxOccurs="1" name="ErrorDescription"
      type="s:string" />
  </s:sequence>
</s:complexType>
<s:element name="ConvertFile2">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="InputPath"
        type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="Options"
        type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="Metadata"
        type="s:base64Binary" />
      <s:element minOccurs="0" maxOccurs="1" name="OutputPath"
        type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="ConvertFile2Response">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="ConvertFile2Result"
        type="tns:ConversionResult" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="ConvertFileWithData">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="DocBytes"
        type="s:base64Binary" />
      <s:element minOccurs="0" maxOccurs="1" name="Options"
        type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="Metadata"
        type="s:base64Binary" />
      <s:element minOccurs="0" maxOccurs="1" name="ParameterName"
        type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="ParameterData"
        type="s:base64Binary" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="ConvertFileWithDataResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1"
        name="ConvertFileWithDataResult"

```

```

        type="tns:ConversionResult" />
    </s:sequence>
</s:complexType>
</s:element>
<s:element name="ConvertUrl">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="Url"
                type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="Options"
                type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="MetadataUrl"
                type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="ConvertUrlResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="ConvertUrlResult"
                type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
</s:schema>
</wsdl:types>
<wsdl:message name="ConvertFileSoapIn">
    <wsdl:part name="parameters" element="tns:ConvertFile" />
</wsdl:message>
<wsdl:message name="ConvertFileSoapOut">
    <wsdl:part name="parameters" element="tns:ConvertFileResponse" />
</wsdl:message>
<wsdl:message name="ConvertFile2SoapIn">
    <wsdl:part name="parameters" element="tns:ConvertFile2" />
</wsdl:message>
<wsdl:message name="ConvertFile2SoapOut">
    <wsdl:part name="parameters" element="tns:ConvertFile2Response" />
</wsdl:message>
<wsdl:message name="ConvertFileWithDataSoapIn">
    <wsdl:part name="parameters" element="tns:ConvertFileWithData" />
</wsdl:message>
<wsdl:message name="ConvertFileWithDataSoapOut">
    <wsdl:part name="parameters" element="tns:ConvertFileWithDataResponse" />
</wsdl:message>
<wsdl:message name="ConvertUrlSoapIn">
    <wsdl:part name="parameters" element="tns:ConvertUrl" />
</wsdl:message>
<wsdl:message name="ConvertUrlSoapOut">
    <wsdl:part name="parameters" element="tns:ConvertUrlResponse" />
</wsdl:message>
<wsdl:portType name="ConverterSoap">
    <wsdl:operation name="ConvertFile">
        <wsdl:input message="tns:ConvertFileSoapIn" />
        <wsdl:output message="tns:ConvertFileSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="ConvertFile2">
        <wsdl:input message="tns:ConvertFile2SoapIn" />
        <wsdl:output message="tns:ConvertFile2SoapOut" />
    </wsdl:operation>

```

```

</wsdl:operation>
<wsdl:operation name="ConvertFileWithData">
  <wsdl:input message="tns:ConvertFileWithDataSoapIn" />
  <wsdl:output message="tns:ConvertFileWithDataSoapOut" />
</wsdl:operation>
<wsdl:operation name="ConvertUrl">
  <wsdl:input message="tns:ConvertUrlSoapIn" />
  <wsdl:output message="tns:ConvertUrlSoapOut" />
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ConverterSoap" type="tns:ConverterSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="ConvertFile">
    <soap:operation soapAction="http://pdf-tools.com/ws/o2p/ConvertFile"
      style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ConvertFile2">
    <soap:operation soapAction="http://pdf-tools.com/ws/o2p/ConvertFile2"
      style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ConvertFileWithData">
    <soap:operation soapAction="http://pdf-tools.com/ws/o2p/
      ConvertFileWithData" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ConvertUrl">
    <soap:operation soapAction="http://pdf-tools.com/ws/o2p/ConvertUrl"
      style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="ConverterSoap12" type="tns:ConverterSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="ConvertFile">
    <soap12:operation soapAction="http://pdf-tools.com/ws/o2p/ConvertFile"
      style="document" />

```

```

    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ConvertFile2">
    <soap12:operation soapAction="http://pdf-tools.com/ws/o2p/ConvertFile2"
      style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ConvertFileWithData">
    <soap12:operation soapAction="http://pdf-tools.com/ws/o2p/
      ConvertFileWithData" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ConvertUrl">
    <soap12:operation soapAction="http://pdf-tools.com/ws/o2p/ConvertUrl"
      style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="Converter">
  <wsdl:port name="ConverterSoap" binding="tns:ConverterSoap">
    <soap:address location="http://dcvs/o2p/Converter.asmx" />
  </wsdl:port>
  <wsdl:port name="ConverterSoap12" binding="tns:ConverterSoap12">
    <soap12:address location="http://dcvs/o2p/Converter.asmx" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

## Web.config

The web.config allows you to adjust settings according to your preferences.

**Note:** The installer makes some adjustments to reflect the installation and log file location (as indicated by the place holder [INSTALLDIR]).

```

<?xml version="1.0"?>
<configuration>
  <appSettings>
    <!--
      Settings used by the web service' ConvertUrl method.
      - ResultLifeTime: the number of seconds to keep the conversion
        results available for download via the returned URL
        Default: 30 seconds.
      - TempDirectory: a path to a file system folder where the method
        shall store temporary files.
        If not specified, the default temporary folder is used.
      - ExpireAfterDownload: the number of seconds to keep the
        conversion result after the download is performed.
        If not specified, the expiration according to
        'ResultLifeTime' remains in force.
      - LogFile: full path and name of log file. Place holder
        "DATE" will be replaced by current date. Empty or missing
        value: no logging.
      - LogLevel: 1=debug, 2=info, 3=error, 4=no logging
    -->
    <add key="ResultLifeTime" value="60"/>
    <add key="TempDirectory" value="[INSTALLDIR]Temp"/>
    <add key="ExpireAfterDownload" value="0"/>
    <add key="LogFile" value="[INSTALLDIR]log\webService-DATE.log"/>
    <add key="LogLevel" value="3"/>
  </appSettings>
  <connectionStrings/>
  <system.webServer>
    <security>
      <requestFiltering>
        <!--
          maxAllowedContentLength limits the maximum size of the
          request content in Bytes. 41943040 corresponds to 40 MB
        -->
        <requestLimits maxAllowedContentLength="41943040"/>
      </requestFiltering>
    </security>
  </system.webServer>
  <system.web>
    <!--
      maxRequestLength limits the maximum size for the request packet
      in kB. 40960 corresponds to 40MB. executionTimeout sets the
      timeout for an HTTP request (600 seconds = 10 minutes)
    -->
    <httpRuntime maxRequestLength="40960" executionTimeout="600"/>
    <!--
      Set compilation debug="true" to insert debugging
      symbols into the compiled page. Because this
      affects performance, set this value to true only
      during development.
    -->
    <compilation debug="false" tempDirectory="[INSTALLDIR]Temp"/>
    <!-- compilation debug="true" targetFramework="4.0"/ -->
    <!--
      The <authentication> section enables configuration
      of the security authentication mode used by

```

```

        ASP.NET to identify an incoming user.
    -->
    <authentication mode="Windows"/>
    <!--
        The <customErrors> section enables configuration
        of what to do if/when an unhandled error occurs
        during the execution of a request. Specifically,
        it enables developers to configure html error pages
        to be displayed in place of a error stack trace.

        <customErrors mode="RemoteOnly"
            defaultRedirect="GenericErrorPage.htm">
            <error statusCode="403" redirect="NoAccess.htm" />
            <error statusCode="404" redirect="FileNotFound.htm" />
        </customErrors>
    -->
    <pages controlRenderingCompatibilityVersion="3.5"
        clientIDMode="AutoID"/>
</system.web>
</configuration>

```

## A.7 Terminalserver

### A.7.1 Platform requirements

The Windows Server platform requires special care with respect to configuring terminal services, since the Document Converter makes use of terminal server sessions for controlling office applications.

When preparing the Windows Server platform, a decision has to be made whether the Terminal Services role shall be installed. Initially, this role is not installed, and the server runs in **administrative mode**. Microsoft has chosen to impose several functional restrictions in this mode compared to the full featured mode with Terminal Service role (**application mode**) enabled:

- the "Start Program" parameter is ignored when starting an RDP session
- the number of connected RDP sessions is limited to two

The Document Converter has some built-in work-around features for these limitations. Notably, it is able to disconnect an RDP session that it has just started, permitting a next RDP session to be initiated.

However, this can be an issue depending on Terminal Services policies in force in certain environments - for instance if disconnected sessions are not tolerated and automatically terminated. Under such conditions, enabling the Terminal Services role will allow to leave multiple sessions connected.

From a licensing point of view, enabling the Terminal Services role means that

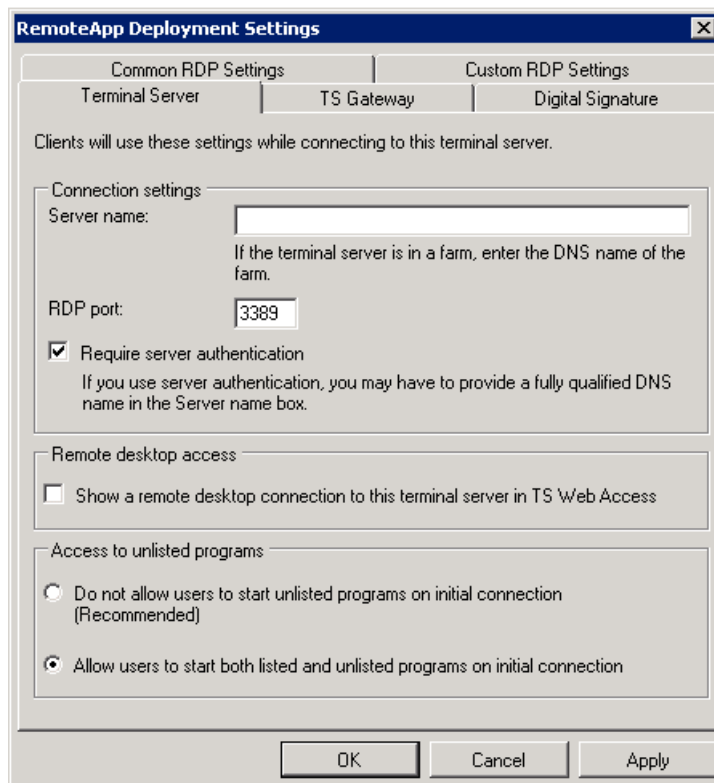
- Remote Desktop Client Access Licenses (RDP CALs) need to be purchased and installed
- MS Office needs to be licensed with a volume license

For Microsoft licensing questions, please refer to the appropriate documentation on the Microsoft websites.

When operating in application mode, make sure the security settings are configured appropriately to

- permit connections appropriately (e.g. do not require client certificates)
- permit starting the work session control program O2PWSC.exe

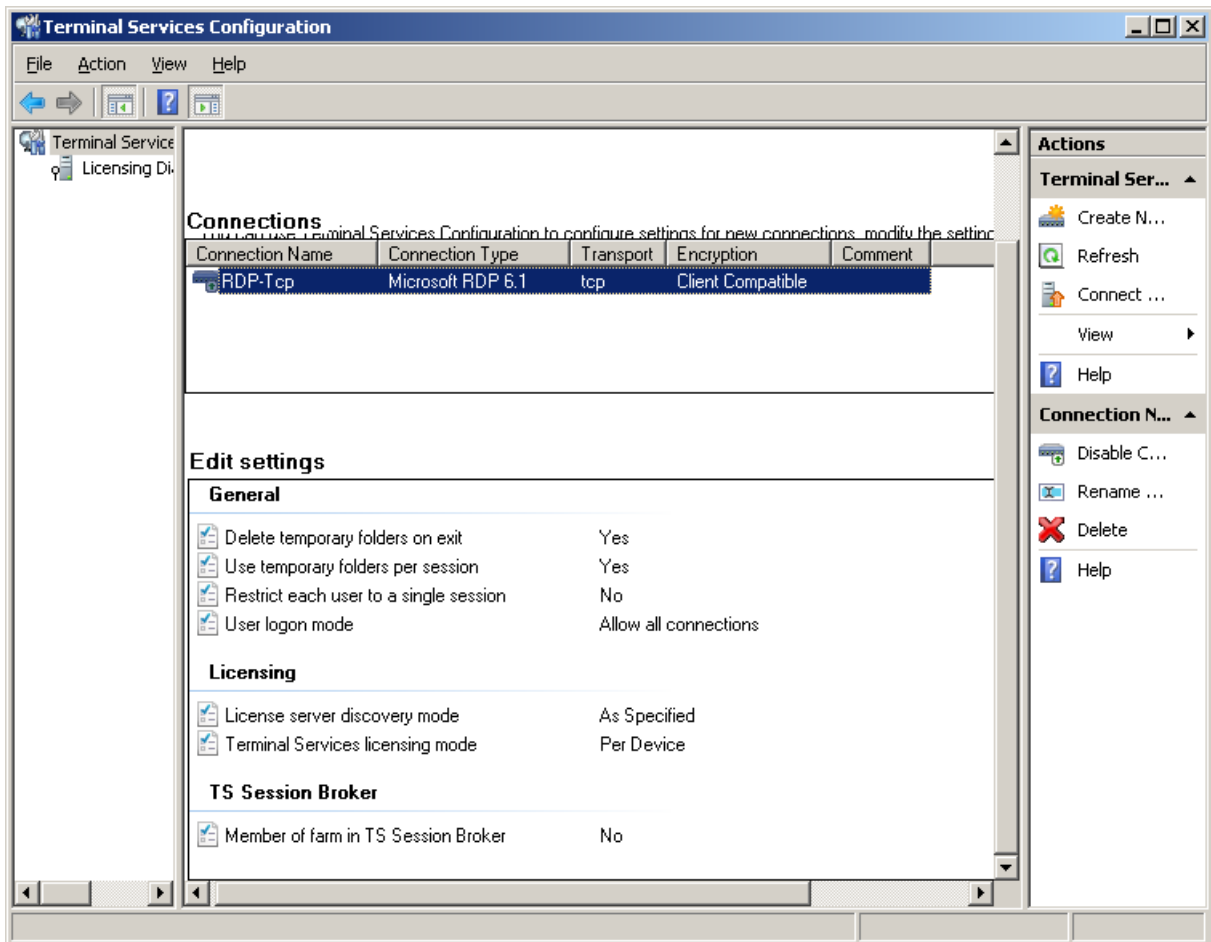
Configurations can be made using the Terminal Server RemoteApp Console (RDP Settings, Terminal Server tab). See figure below.



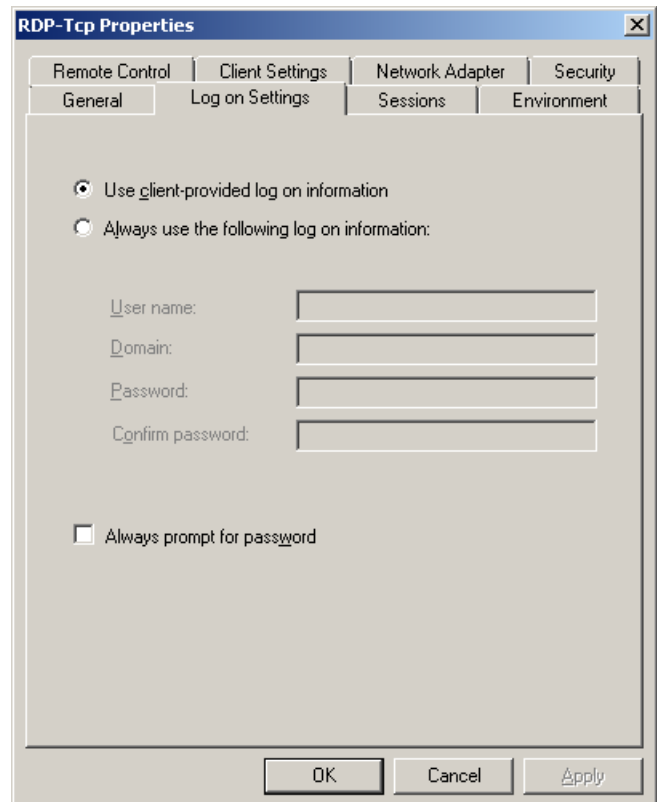
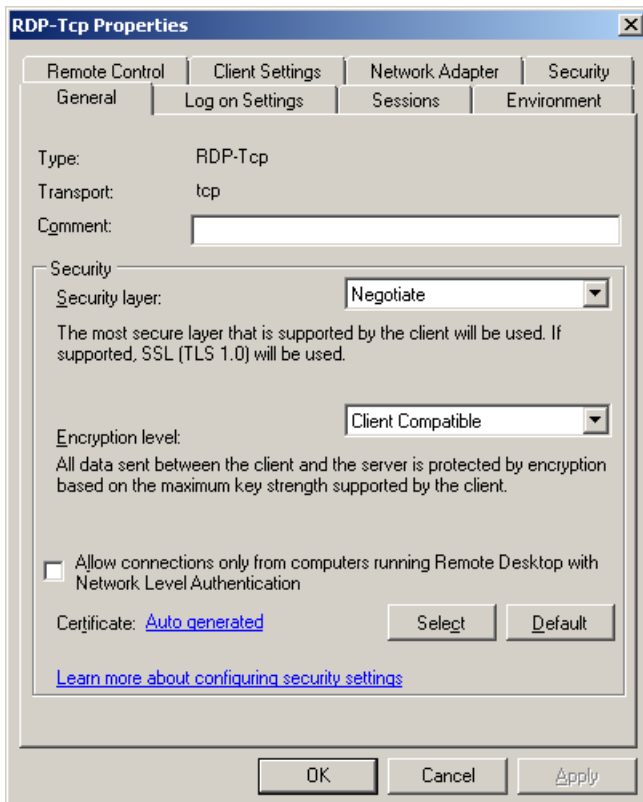
## A.7.2 Configuration of Terminal Services on Windows 2008 Server

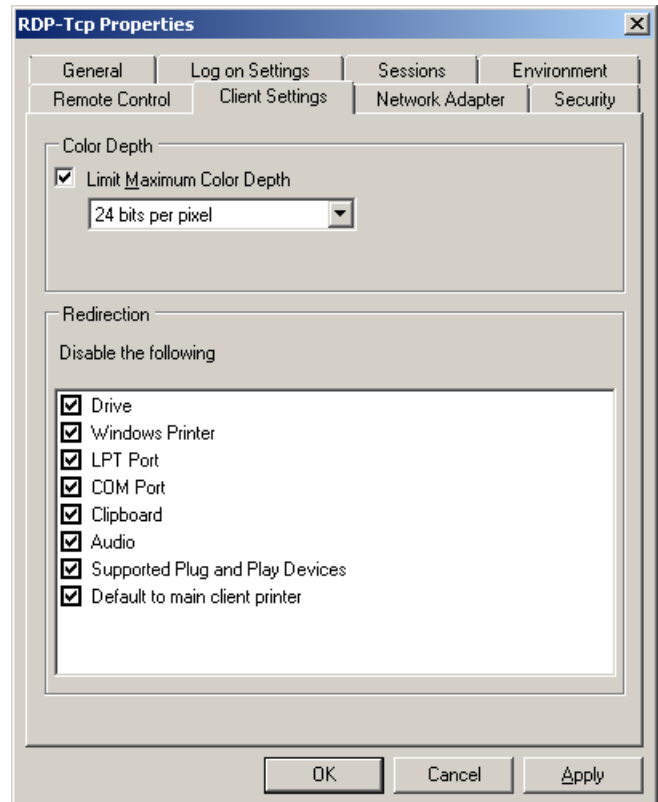
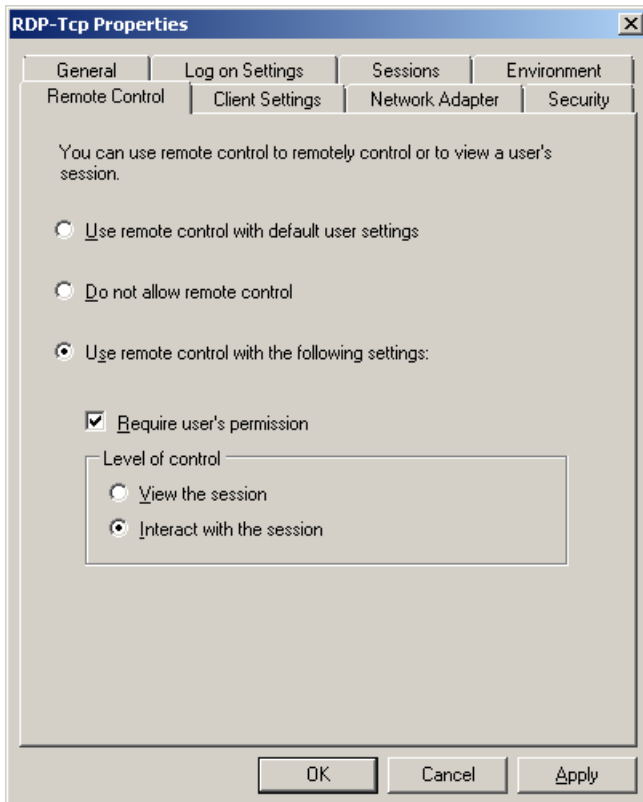
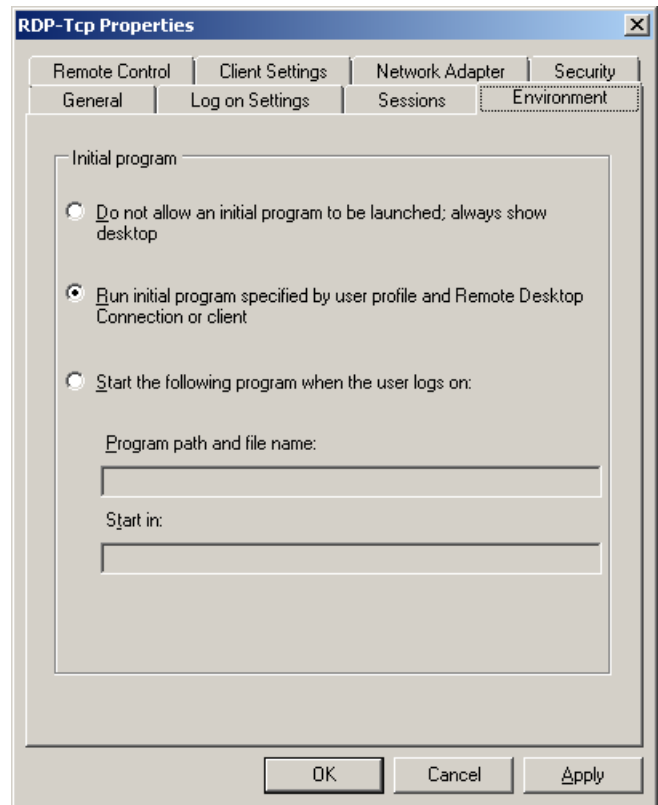
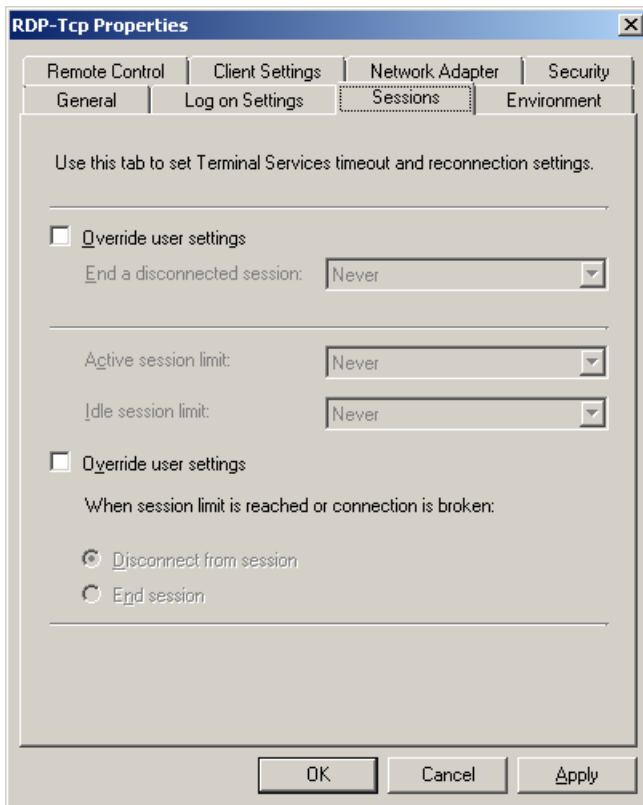
Go through the following configuration dialogues to adjust the settings.

Start with "Terminal Services Configuration" (Start → Administrative Tools → Terminal Services):

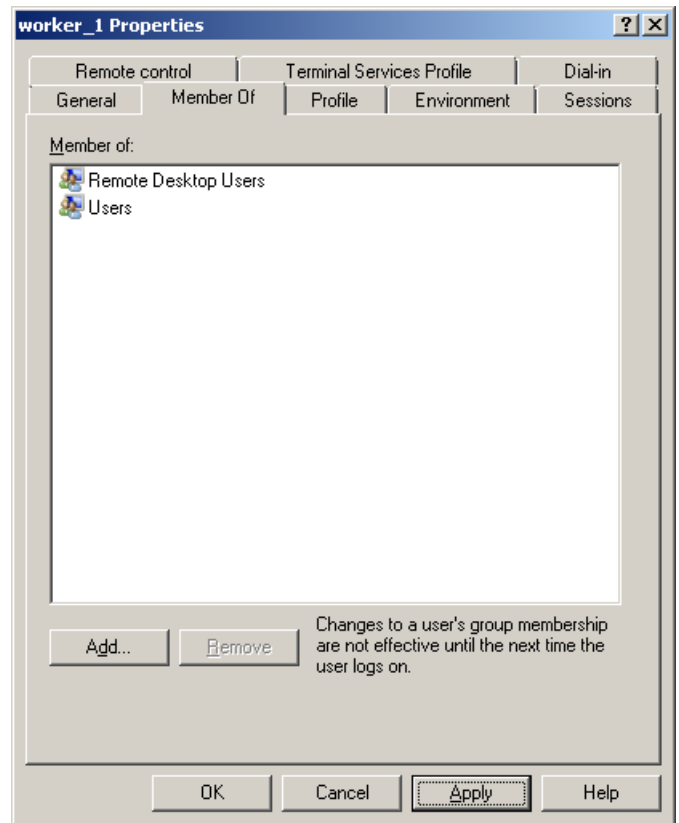
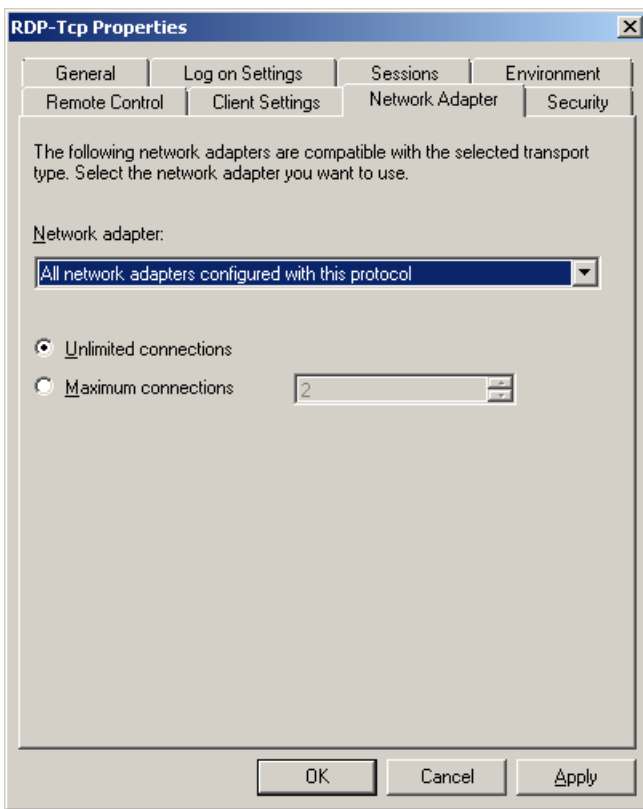


With a right click on the "RDP-Tcp" connection line (selected above), open the properties dialogue and go through the tabs as shown on the subsequent screen shots:

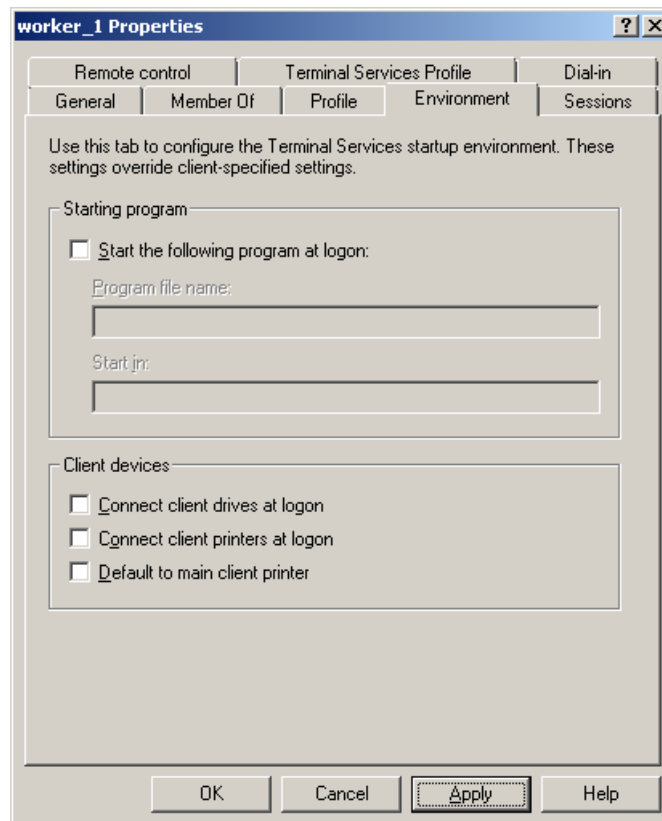




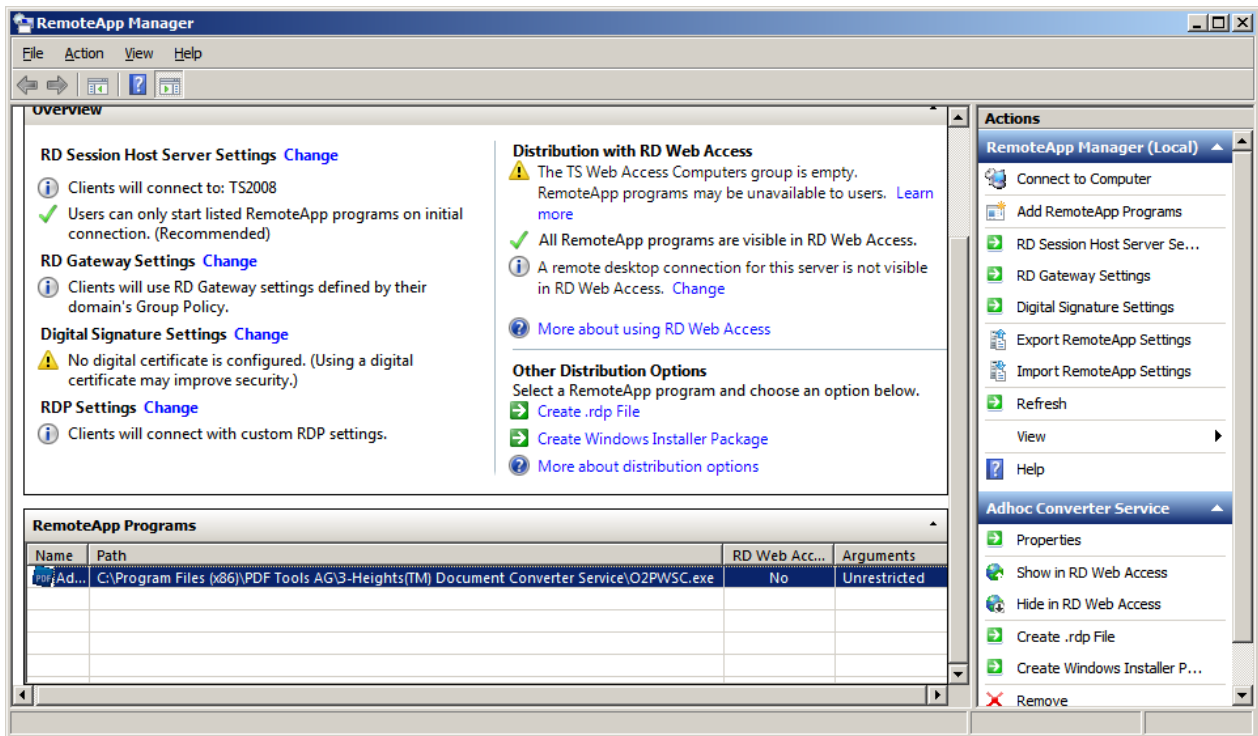
The Color Depth setting is only relevant if you intend to use the conversion of web pages via Internet Explorer. Otherwise, you can leave the default 16 bits per pixel.



You need to create and configure one or several user accounts for use with the Document Converter. Make sure these accounts have sufficient rights to log on via Remote Desktop.



When Remote Desktop Services are configured in application mode, start "RemoteApp" manager and enter 02PWSC.exe as program that is allowed to be started with any parameter value (see figure below).



Finally, log in interactively to each account created, and start each office application to go through any initial application setup dialogues.

For more information about configuring Terminal Services, please refer to the [Microsoft documentation](#).

### A.7.3 Troubleshooting

**Worker sessions are started, but the work session control program does not start automatically** There are several policies and registry settings which can possibly interfere with the automatic session start via RDP. Please refer to section [Windows policies and registry settings](#)

## A.8 Authorization of Exchange Online access

For modern authentication to be used for accessing Exchange Online mail folders, an appropriate configuration must be prepared in Azure Active Directory:

- Open "[App registrations](#)" in Azure Active Directory
- click "New registration" and enter a name (e.g. Exchangelmap), select "Public client/native (mobile & desktop)", then click "Register"
- click "API permissions" and add the following permissions:
  - Exchange: Mail.Read Delegated (look for Exchange in Microsoft APIs at the bottom of the page, click and choose Delegated permissions)
  - Microsoft Graph (delegated): User.Read will already be selected; also add the following items:
    - IMAP.AccessAsUser.All
    - Mail.Read
    - Mail.ReadWrite
  - click "Grant admin consent for ..." and confirm
- click "Authentication" (left pane); at the bottom of the page, toggle "Treat application as public client" to "Yes" and "Save"

When configuring an Exchange Online folder for the Document Converter, you will need to provide the Tenant ID as well as the Application (client) ID.

The screenshot shows a dialog box titled "Location Details" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- Type:** A dropdown menu set to "Exchange Online Folder".
- Tenant ID:** A text input field containing the GUID "8ef7c4e2-6dae-4e58-a1ae-8aa620dc1c76".
- Client ID:** A text input field containing the GUID "632c6812-576c-4f35-8946-fcd2eadd52ec".
- User Account:** A text input field containing "user.name@domain.com".
- Password:** A text input field with masked characters "\*\*\*\*\*".
- Password (re-enter):** A text input field with masked characters "\*\*\*\*\*".
- Folder:** A dropdown menu set to "INBOX".
- refresh:** A button located to the right of the Folder dropdown.
- Cancel:** A button at the bottom center.
- OK:** A button at the bottom right, highlighted with a blue border.