

Introduction

Several command line utilities for converting files and folders are included with Document Conversion Service. These utilities can be called from scheduled tasks, batch files or any program that can call an external program.

Utility	Descriptions
DCSConvertFile	Converts a file using Document Conversion Service. The file can be converted in place or saved in a different location.
DCSConvertFileList	Given a text file containing a list of files to convert, or a list of files provided on the command line, converts all files using Document Conversion Service.
DCSConvertFolder	Walks the given folder and converts all files, or all files matching a provided search filter, using Document Conversion Service. The utility can optionally also process all subfolders under the starting folder as well.
DCSCombineFiles	Given a text file containing a list of files, and/or a list of files provided on the command line, this utility will convert the files using Document Conversion Service and combine all of the files together into a single multipaged file or a collection of serialized pages. The files are appended together in the order in which they are received.
DCSCombineFolder	Walks a folder and combines all files, or all files matching a search filter, together into a single file or a collection of serialized pages using Document Conversion Service. The utility can optionally also process all subfolders under the starting folder as well.
DCSExtractResults	Each of the above utilities can create a results log file containing a complete snapshot of the conversion information for each file converted, in both a success and failure case. This utility can be used to extract information from the results log files, such as all files created, or any errors that occurred.
DCSCreateFileList	Searches a folder, and optionally any subfolders and lists of files matching the search filter specified.
DCSLicenseDaysLeft	Echos to the command line the current license level and how many days remain in the subscription.

When using the above command line utilities the type of output created is based on the conversion profile passed. A selection of common conversion profiles are included with the Document Conversion Service install. See the section [Conversion Settings](#) for more details, and instructions on [creating your own custom profiles](#).

The command line utilities all return the following error codes:

- 0 – success
- 1 – failed
- 2 – invalid parameters, when the command line tool has parameters

The results of each command line utility are sent to standard out, while any errors that may have happened are sent to standard error. To capture this information you can:

- use the output redirection operators `>` and `>>` to save or append the standard output results in a file
- use the pipe operator `(|)` to send the standard output to another program as input
- redirect only the standard output to a file with the operator `1>C:\DCS\files.txt`
- redirect only the standard error to a file with the operator `2>C:\DCS\err.txt`

DCSConvertFile

A command line utility to convert a file using Document Conversion Service. The Document Conversion Service must be running, either locally or on a remote computer for the file to be converted. If it is not running the command will return immediately with an error.

```
DCSConvertFile /P=profile [/S=save location] [/N=output name] [/O] [/NE] [/L]
[/D="name:value"] [/E=extension map]
[/W=wait time] [/FAIL=failed results log file location]
[/SIL=conversion log file path]
[/C=remote computer name;remote scratch folder]
[/T=alternate temp folder]
sourcefile
```

Sample Command Lines

Convert a single file to a TIFF:

```
DCSConvertFile /P="TIFF 200dpi Monochrome" "C:\Test\Document.doc"
```

Send the file C:\Test\Document.doc to Document Conversion Service to be converted using the settings contained in the conversion profile *TIFF 200dpi Monochrome.xml*.

The converted file, *Document.doc.tif*, is saved in C:\Test\, the same location as the source file.

If a file of the same name already existed, this conversion would fail and a *.failed* folder would be created in the same location as the source document, C:\Test. The results log would be named *Document.doc.failed.dcsresults* and saved to a new subfolder under the *.failed* folder. The subfolder is named using the date and time of the conversion to keep subsequent runs separate.

To overwrite an existing file the */O* switch would need to be added to the above command. If you did not want the source file extension as part of your file name, the */NE* switch would need to be added.

Convert a single file to a TIFF with a specific output name, overwrite existing files:

```
DCSConvertFile /O /N="Opt_Document" /P="TIFF 300dpi OptimizedColor"
"C:\Test\Document.doc"
```

Send the file C:\Test\Document.doc to Document Conversion Service to be converted using the settings contained in the conversion profile *TIFF 300dpi OptimizedColor.xml*.

The converted file will be named *Opt_Document.tif* and saved in the same location as the source file. If a file of the same name already existed, this file would be overwritten with the new file.

Convert a single file to a TIFF in a specific location:

```
DCSConvertFile /S="C:\Output" /P="TIFF 200dpi Monochrome" "C:\Test\Document.doc"
```

Send the file C:\Test\Document.doc to Document Conversion Service to be converted using the settings contained in the conversion profile *TIFF 200dpi Monochrome.xml*.

The converted file will be named *Document.doc.tif* and placed in the folder named C:\Output.

If a file of the same name already existed, this conversion would fail and a *.failed* folder would be created in the same location as the source document, C:\Test. The results log would be named *Document.doc.failed.dcsresults* and saved to a new subfolder under the *.failed* folder. The subfolder is named using the date and time of the conversion to keep subsequent runs separate.

Convert a single file to a TIFF in a specific location, wait up to 5 minutes for the conversion service to start:

```
DCSConvertFile /S="C:\Output" /P="TIFF 200dpi Monochrome" /W=300
"C:\Test\Document.doc"
```

Send the file C:\Test\Document.doc to Document Conversion Service to be converted using the settings contained in the conversion profile *TIFF 200dpi Monochrome.xml*. If Document Conversion Service is not running, wait up to 5 minutes (300 seconds) for the conversion service to be available.

The converted file will be named *Document.doc.tif* and placed in the folder named C:\Output.

If a file of the same name already existed, this conversion would fail and a *.failed* folder would be created in the same location as the source document, C:\Test. The results log would be named *Document.doc.failed.dcsresults* and saved to a new subfolder under the *.failed* folder. The subfolder is named using the date and time of the conversion to keep subsequent runs separate.

Convert a single file to a vector PDF document, remove source extension and save the conversion results log file:

```
DCSConvertFile /P="Adobe PDF Multipage" /L /NE "C:\Test\Document.doc"
/FAIL="C:\Test\FailedLogs\\" /D="UseDateTimeInFailedFolder:FALSE"
```

Send the file C:\Test\Document.doc to Document Conversion Service to be converted using the settings contained in the conversion profile *Adobe PDF Multipage.xml*. This profile creates vector PDF where possible.

The converted file will be named *Document.doc.pdf* and saved in the same location as the source file. A conversion results log file, *Document.doc.succeeded.dcsresults* will also be saved in same location as the source file.

If a file of the same name already existed, this conversion would fail. The results log would be named *Document.doc.failed.dcsresults* and would be saved in the C:\Test\FailedLogs\ folder. The /D setting *UseDateTimeInFailedFolder* disables the date and time subfolder creation under the failed logs folder.

Note: The double ending backslash used when specifying the folder for the /FAIL switch is required for the command line path to be parsed correctly.

To overwrite an existing file the /O flag would need to be added to the above command.

Convert a single file to a vector PDF document, save the output to a specific location and save the conversion results log file:

```
DCSConvertFile /S="C:\Output" /N="NewFileName" /L /O
/P="Adobe PDF Multipage" "C:\Test\Document.doc"
```

Creates a PDF document from the source file C:\Test\Document.doc. The type of PDF created is controlled by the settings in the conversion profile *Adobe PDF Multipage.xml*. This profile creates vector PDF where possible.

The converted file will be named *NewFileName.pdf* and saved in the C:\Output folder. If a file of the same name already exists in C:\Output\ this file would be overwritten with the new file.

A conversion results log file, *Document.doc.succeeded.dcsresults* will also be created in the C:\Output folder.

If the conversion did not succeed, the results log would be named *Document.doc.failed.dcsresults* and a *.failed* folder would be created in the same location as the source document, C:\Test. The results log would be named *Document.doc.failed.dcsresults* and saved to a new subfolder under the *.failed* folder. The subfolder is named using the date and time of the conversion to keep subsequent runs separate.

Convert a single file to a PDF document, save both the output and any failed conversion results log file to custom locations :

```
DSCSConvertFile /S="C:\Output" /N="NewFileName" /L /O  
/P="PDF A-1b 300dpi OptimizedColor"  
/FAIL="C:\FailedResults\\" "C:\Test\Document.doc"
```

Creates a PDF document from the source file `C:\Test\Document.doc`. The type of PDF created is controlled by the settings in the conversion profile *PDF A-1b 300dpi OptimizedColor.xml*.

The converted file will be named *NewFileName.pdf* and saved in the `C:\Output` folder. If a file of the same name already exists in `C:\Output\` this file would be overwritten with the new file as specified by the `/O` argument.

A conversion results log file, *Document.doc.succeeded.dcsresults* will be created in the `C:\FailedResults` folder if the conversion does not succeed. The results log is named *Document.doc.failed.dcsresults* and placed in a subfolder named with the current date and time created under the specified folder.

Note: The double ending backslash used when specifying the folder for the `/FAIL` switch is required for the command line path to be parsed correctly.

Use the command line argument `D="UseDateTimeInFailedFolder:FALSE"` to store the results log file directly in the folder `C:\FailedResults`.

Command Line Arguments

Command line switches are not case-sensitive and can be entered in either upper or lower case.

/S - The Save Location

Pass in the full path to the folder in which to save the new file. If the save location is not specified the new file is created in the same folder as the source file.

- If the path includes spaces it must be enclosed in quotes.
- If the path doesn't exist, the conversion will fail.
- If a file of the same name already exists in the save file location, the conversion will fail. The /O option can be used to enable file overwriting, which is off by default.

Example:

```
/S="C:\Converted Files\Test"
```

/N - Output File Name

The name to use for the output file. The default file extension for the type of file being created will always be added to the name provided here.

If this argument is not specified the name of the source file, including the extension is used. This prevents name collision when you have two different files with the same base name, such as abc.doc and abc.pdf. If you were converting to multipaged TIFF images the resulting converted files would be named abc.doc.tif and abc.pdf.tif.

If you do not want the original file name extension as part of your file name, use the /NE switch to remove the file extension.

If serialized files, such as JPEG images, are being created, the base name will be appended with the page number, SampleDocument_0001.jpg, SampleDocument_002.jpg, etc.

Example:

```
/N="SampleDocument_06_15_2012"
```

/O - Overwrite Always

Enables overwrite mode so that existing files of the same name are overwritten with the new file. When not specified the conversion will fail if a file of the same name already exists in the output folder.

/NE - No Extension

If you do not want the original file name extension as part of your output file name, use this switch to remove the file extension. If you have provided an output name with the /N switch above, this argument is ignored.

/L - Results Log

The results log file is an XML file containing a complete snapshot of the conversion information. Normally only saved for failed conversions, the /L argument enables creation of the results log file when the conversion succeeds.

The name of the results log file is based on the name of the original file and also indicates the conversion status. For example, when converting *Document.doc*, a successful conversion will create a log file named *Document.doc.succeeded.dcsresults.*, while a failed conversion would be named *Document.doc.failed.dcsresults.*

The results log file for a successful conversion is always copied to the output location with the converted files when this flag is used.

In the case of a failed conversion, the log file is always created. See the /FAIL switch to control the location and creation of the failed results log files.

The result log files can later be passed to the [DCSExtractResults](#) command line utility to extract information such as all files created or any errors encountered during conversion. You can search a folder for the results log files using the [DCSCreateFileList](#) utility.

/FAIL - Failed Results Log File Location

In the case of a failed conversion, the conversion results log file is always created. The default behavior is to create a *.failed* folder in the same location as the source file and save the conversion results log file to a new subfolder under the *.failed* folder. The subfolder is named using the date and time of the conversion to keep subsequent runs separate.

This argument allows you to override the default use of the *.failed* folder and to provide a specific folder in which to store the failed results log files. The name of the results log file is based on the name of the original file and also indicates the conversion status. For example, when converting *Document.doc*, a failed conversion would be named *Document.doc.failed.dcsresults.*

You can suppress the use of the date and time subfolder by passing the *UseDateTimeInFailedFolder* setting using the /D switch.

If you do not want to create the failed results log files at all, you can use the /D switch to pass the *KeepFailedItemResultsFiles* setting as *false*. These settings can also be added to any conversion profile you are using.

The result log files can later be passed to the [DCSExtractResults](#) command line utility to extract information such the source file used or any errors encountered during conversion. You can search a folder for the results log files using the [DCSCreateFileList](#) utility.

Note: The double ending backslash used when specifying the folder for the /FAIL switch is required for the command line path to be parsed correctly.

Examples:

```
/FAIL="C:\ConvertedFiles\Failed\\" /D="UseDateTimeInFailedFolder:FALSE"
```

/P - Conversion Profile

This is a required argument. The type of file created is controlled by supplying a conversion profile using this switch. The profiles are referenced by passing in the name of the profile XML file, with or without the XML extension. See [Creating and Customizing Profiles](#) for more information about the contents of the profiles, a list of profiles included with Document Conversion Service, and how to create your own.

Examples:

```
/P="TIFF 300dpi Color Fax"  
/P="TIFF 204x196dpi Monochrome Fax.xml"
```

/D - Define Setting

Individual conversion and profile settings can be supplied on the command line using this switch. This switch can be specified multiple times for separate settings and any settings passed here will override the settings in the profile.

Any name-value pair that can be written in a profile can be passed through this parameter. This includes options to control the conversion settings as well as the behavior of the individual converters as well. See [Creating and Customizing Profiles](#) for more information about the name-value pairs that can be used.

Examples:

These first two are settings that control the converter options, such as what pages to print, and the output that PowerPoint will print.

```
/D="PrintRange:1-5"  
/D="PowerPoint.PrintOptionsOutputType:PrintOutputNotesPages"
```

These two settings control the output file creation options, and would override or add to the settings in the conversion profile passed using the /P switch.

```
/D="Image Options;Fax Resolution:3"  
/D="TIFF File Format;BW compression:Group3-2D"
```

These two settings control the where the failed results log files are created and are most often used along with the /FAIL switch to control where the results log files are saved.

```
/D="KeepFailedItemResultsFiles:TRUE"  
/D="UseDateTimeInFailedFolder:FALSE"
```

/E - File Extension Mapping

A file extension mapping profile uses the extension of the source file to determine what converter will be used to convert the file. Like the conversion profiles, this file is also an XML file. This switch is optional and an internal default mapping is provided. You would only need to provide this file if you wanted to override the default file extension to converter mappings provided.

Examples:

```
/E="Custom Extension To Converter Map"
```

/W - Wait Time

Use this switch to wait to the specified number of seconds for the Document Conversion Service to be running and available to convert documents. If Document Conversion Service is already running the command executes immediately. If the Document Conversion Service is not running in the timeout period specified, the command will return with an error.

If this argument is not specified the command will return immediately with an error if Document Conversion Service is not running.

Example:

```
/W=300
```

/C - Convert on a Remote Computer (DCOM)

If Document Conversion Service is running on a different computer, use this switch to pass the name of the remote computer and the path of a shared location that both computers have access to. Separate the name of the remote computer and the path to the shared folder location with a semi-colon.

When converting remotely, the client redistributable, PNDocConvClientSetup_3.0.exe, must be installed on the computer running this command line utility. The client setup install program is included as part of the Document Conversion Service install and can be found in the **\Samples\Redist** folder in your product installation folder.

Examples:

```
/C="DOCCONV_SERVER;\\DOCCONV_SERVER\DCSREMOTE"
```

/SIL - Smart Inspect Logging File

Smart Inspect Log files are a tracing of the entire conversion process and are not the same as the conversion results log files created when a conversion fails. These logs can be viewed using the SmartInspect Redistributable Console included with Document Conversion Service.

These log files are automatically deleted when conversion succeeds. To keep the log files on success use the custom setting *AlwaysKeepProcessingLoggingFiles* as shown below.

The default location for this file is the TEMP folder. Each logging file is assigned a unique date, time and thread prefix followed by "_PNConvertFile.sil", such as *2014_09_11_2_38_00_PM_4_PNConvertFile.sil*.

Use this argument to specify a custom path and optional file name for the SmartInspect logging file (*.sil) created by this utility. The /SIL switch can take a folder, or a path to a filename. If a path without a trailing backslash is provided, the last part of the path is assumed to be a filename.

Note: The double ending backslash used when specifying a folder for the /SIL switch is required for the command line path to be parsed correctly.

/SIL=	Is interpreted as...
"C:\Test\LogFile"	Create the SmartInspect log file as C:\Test\LogFile.sil.
"C:\Test\LogFile\"	Create the SmartInspect log file as C:\Test\LogFile\datetime_PNConvertFile.sil
"C:\Test\LogFile\ConvertFileCustom.sil"	Create the SmartInspect log file as C:\Test\LogFile\ConvertFileCustom.sil

The following settings can be used to control the creation and naming of the logging file. These settings are all passed using the /D switch.

Custom Setting	Description
RemoveDateTimePrefixOnProcessingLoggingFiles	Pass <i>True</i> to disable the adding of the unique date, time and thread prefix when a custom file name has not been specified in the <i>ConvertFileProcessLoggingPath</i> parameter.
KeepFailedProcessingLoggingFiles	Pass as <i>False</i> to disable the automatic creation of SmartInspect logging files when conversion fails. This setting can be overridden by <i>AlwaysKeepProcessingLoggingFiles</i> .
AlwaysKeepProcessingLoggingFiles	When set to <i>True</i> , the SmartInspect logging files are always created in the %TEMP% or other specified folder for both successful and failed conversions. If set to <i>False</i> , no logging files are created. This setting will override the <i>KeepFailedProcessingLoggingFiles</i> setting.

Examples:

Pass a custom folder and remove the prefix, each run will overwrite the log file C:\PEERNET\Logs\PNConvertFile.sil.

```
/SIL="C:\PEERNET\Logs\" /D="RemoveDateTimePrefixOnProcessingLoggingFiles:TRUE"
```

Pass a custom folder and log file name and remove the prefix. Each run will overwrite the logging file C:\PEERNET\Logs\MyLogFile.sil.

```
/SIL="C:\PEERNET\Logs\MyLogFile" /D="RemoveDateTimePrefixOnProcessingLoggingFiles:TRUE"
```

Don't save any SmartInspect log files at all.

```
/D="AlwaysKeepProcessingLoggingFiles:FALSE"
```


/T - Alternate Temp Folder

This is an advanced setting that should not be needed in most cases. When converting a file, the conversion tool copies the file and performs the conversion in temporary *staging* and *working* folders created on demand in the default Windows temp folder. When dealing with long path and file names the default folders created can occasionally cause path names that are too long to process. When this happens this switch can be used to set the temporary folder to a shorter path to allow processing.

This setting is overridden if the /C option for remote conversion is being used with its own path to a shared location for conversion.

Examples:

```
/T= "C:\PNTemp\\"
```

/? - Display Help

When passed as the only argument this switch will display help for this command.

Source File

The full path to the file to convert.

- If the path to the file includes spaces it must be enclosed in quotes.
- If the file doesn't exist, the conversion will fail.

DCSConvertFileList

A command line that accepts a text file containing a list of files to convert, or a list of files provided on the command line, and converts all files using Document Conversion Service. The Document Conversion Service must be running, either locally or on a remote computer for the file to be converted. If it is not running the command will return immediately with an error.

```
DCSConvertFileList /P=profile [/S=save location] [/O] [/NE] [/L]
                  [/E=extension map]
                  [/C=remote computer name;remote scratch folder]
                  [/D="name:value"] [/W=wait time]
                  [/FAIL=failed results log file location] [/SIL=conversion log file path]
                  [/I=input text file path]
                  [/T=alternate temp folder]
                  "file[:save location]" "file[:save location]"...
```

Sample Command Lines

Convert all files on command line to TIFF images:

```
DCSConvertFileList /P="TIFF 200dpi Monochrome.xml"
                  "C:\Input\File1.doc" "C:\Input2\File2.doc"
```

Sends the files C:\Input\File1.doc and C:\Input\File2.doc to Document Conversion Service to be converted using the settings contained in the conversion profile *TIFF 200dpi Monochrome.xml*.

The converted files, *File1.doc.tif* and *File2.doc.tif*, will each be saved in the same location as their source file.

If a file with the same name already exists, that file conversion would fail. The results log file, named based on the source file and ending with *.doc.failed.dcsresults* would be placed in a folder named *.failed* created in the same location as the source document. This can be controlled with the */FAIL* switch.

To overwrite an existing file the */O* switch would need to be added to the above command. If you did not want the source file extension as part of your file name, the */NE* switch would need to be added.

Convert all files on command line to TIFF images in their own directory:

```
DCSConvertFileList /P="TIFF 200dpi Monochrome.xml" "C:\Input\File1.doc;C:\Output1"
                  "C:\Input2\File2.doc;C:\Output2"
```

Sends the files C:\Input\File1.doc and C:\Input\File2.doc to Document Conversion Service to be converted using the settings contained in the conversion profile *TIFF 200dpi Monochrome.xml*.

The converted file, *File1.doc.tif* will be saved to the directory *C:\Output1* and *File2.doc.tif* will be saved in the directory *C:\Output2*.

If the output directory does not exist, or if a file with the same name already exists in either directory, that file conversion will fail. The results log file, named based on the source file and ending with *.doc.failed.dcsresults* would be placed in a folder named *.failed* created in the same location as each source document. This can be controlled with the */FAIL* switch.

To overwrite an existing file the */O* switch would need to be added to the above command. If you did not want the source file extension as part of your file name, the */NE* switch would need to be added.

Convert all files in the input file to TIFF images in a specific location:

```
DCSConvertFileList /P="TIFF 300dpi OptimizedColor.xml" /S="C:\Test\Output"  
/I="C:\Test\Files.txt"
```

Sends the files listed in the text file *Files.txt* to Document Conversion Service to be converted using the settings contained in the conversion profile *TIFF 300dpi OptimizedColor.xml*.

Upon successful conversion each output file is placed under the C:\Test\Output folder.

If a file with the same name already exists, that file conversion would fail. The results log file, named based on the source file and ending with *.doc.failed.dcsresults* would be placed in a folder named *.failed* created in the same location as the source document.

Convert a list of files to vector PDF, strip off the source extension and save the output and results log files to a specific location:

```
DCSConvertFileList /P="Adobe PDF Multipage.xml" /NE /S="C:\Test\Output" /L  
/FAIL="C:\Test\FailedLogs\\" /D="UseDateTimeInFailedFolder:FALSE"  
/I="C:\Test\Files.txt"
```

Creates a PDF file from each file listed in the input file *Files.txt*. The PDF created is a vector PDF as controlled by the settings in the conversion profile *Adobe PDF Multipage.xml*.

Upon successful conversion each output file is placed under the C:\Test\Output folder along with the conversion results log file. The name of the results log file is based on the original source file and also indicates the conversion status. For example, if the source file name was *SampleDocument.doc*, a results log file, *SampleDocument.doc.succeeded.dcsresults*, will be created if the conversion succeeds.

The */NE* flag causes the output file to be named using the base name of the original file, plus the extension of the file type you are creating. If a file of that name already exists in the folder the conversion will fail. If the conversion fails a results log file named based on the source file and ending with *.failed.dcsresults* is placed into the folder *C:\Test\FailedLogs* specified by the */FAIL* parameter. The */D* setting *UseDateTimeInFailedFolder* disables the date and time subfolder creation under the failed logs folder.

Note: The double ending backslash used when specifying the folder for the */FAIL* switch is required for the command line path to be parsed correctly.

Command Line Arguments

Command line switches are not case-sensitive and can be entered in either upper or lower case.

/S - The Save Location

Pass in the full path to the folder in which to save the new files. If the save location is not specified the new file is created in the same folder as the source file. If the files listed in the input file text file specified with the /I switch also include save locations, those locations will be used instead.

- If the path includes spaces it must be enclosed in quotes.
- If the path doesn't exist, the conversion will fail.
- If a file of the same name already exists in the save file location, the conversion will fail. The /O option can be used to enable file overwriting, which is off by default.

Example:

```
/S="C:\Converted Files\Test"
```

/O - Overwrite Always

Enables overwrite mode so that existing files of the same name are overwritten with the new file. If the overwrite switch is not specified, the conversion of that file in the list of files will fail if a file of the same name already exists in the output folder.

/NE - No Extension

If you do not want the original file name extension as part of your output file name, use this switch to remove the file extension.

/L - Results Log

The results log file is an XML file containing a complete snapshot of the conversion information. Normally only saved for failed conversions, the /L argument enables creation of the results log file when the conversion succeeds. The results log file is placed in the same location as the converted files.

The name of the results log file is based on the name of the original file and also indicates the conversion status. For example, when converting *Document.doc*, a successful conversion will create a log file named *Document.doc.succeeded.dcsresults*, while a failed conversion would be named *Document.doc.failed.dcsresults*.

The results log file for a successful conversion is always copied to the output location with the converted files when this flag is used.

In the case of a failed conversion, the log file is always created. See the /FAIL switch to control the location and creation of the failed results log files.

The result log files can later be passed to the [DCSExtractResults](#) command line utility to extract information such as all files created or any errors encountered during conversion. You can search a folder for the results log files using the [DCSCreateFileList](#) utility.

/FAIL - Failed Results Log File Location

In the case of a failed conversion, the conversion results log file is always created. The default behavior is to create a *.failed* folder in the same location as the source file and save the conversion results log file to a new subfolder under the *.failed* folder. The subfolder is named using the date and time of the conversion to keep subsequent runs separate.

This argument allows you to override the default use of the *.failed* folder and to provide a specific folder in which to store the failed results log files. The name of the results log file is based on the name of the original file and also indicates the conversion status. For example, when converting *Document.doc*, a failed conversion would be named *Document.doc.failed.dcsresults*.

You can suppress the use of the date and time subfolder by passing the *UseDateTimeInFailedFolder* setting using the */D* switch.

Note: The double ending backslash used when specifying the folder for the */FAIL* switch is required for the command line path to be parsed correctly.

Examples:

```
/FAIL="C:\ConvertedFiles\Failed\\" /D="UseDateTimeInFailedFolder:FALSE"
```

If you do not want to create the failed results log files at all, you can use the */D* switch to pass the *KeepFailedItemResultsFiles* setting as *false*.

On the command line:

```
/D="KeepFailedItemResultsFiles:False"
```

In a conversion profile:

```
<add Name="KeepFailedItemResultsFiles" Value="False"/>
```

The result log files can later be passed to the [DCSExtractResults](#) command line utility to extract information such the source file used or any errors encountered during conversion. You can search a folder for the results log files using the [DCSCreateFileList](#) utility.

/P - Conversion Profile

This is a required argument. The type of file created is controlled by supplying a conversion profile using this switch. The profiles are referenced by passing in the name of the profile XML file, with or without the XML extension. See [Creating and Customizing Profiles](#) for more information about the contents of the profiles, a list of profiles included with Document Conversion Service, and how to create your own.

Examples:

```
/P="TIFF 300dpi Color Fax"  
/P="TIFF 204x196dpi Monochrome Fax.xml"
```

/D - Define Setting

Individual conversion and profile settings can be supplied on the command line using this switch. This switch can be specified multiple times for separate settings and any settings passed here will override the settings in the profile.

Any name-value pair that can be written in a profile can be passed through this parameter. This includes options to control the conversion settings as well as the behavior of the individual converters as well. See [Creating and Customizing Profiles](#) for more information about the name-value pairs that can be used.

Examples:

These first two are settings that control the converter options, such as what pages to print, and the output that PowerPoint will print.

```
/D="PrintRange:1-5"
```

```
/D="PowerPoint.PrintOptionsOutputType:PrintOutputNotesPages"
```

These two settings control the output file creation options, and would override or add to the settings in the conversion profile passed using the /P switch.

```
/D="Image Options;Fax Resolution:3"
```

```
/D="TIFF File Format;BW compression:Group3-2D"
```

These two settings control where the failed results log files are created and are most often used along with the /FAIL switch to control where the results log files are saved.

```
/D="KeepFailedItemResultsFiles:TRUE"
```

```
/D="UseDateTimeInFailedFolder:FALSE"
```

/E - File Extension Mapping

A file extension mapping profile uses the extension of the source file to determine what converter will be used to convert the file. Like the conversion profiles, this file is also an XML file. This switch is optional and an internal default mapping is provided. You would only need to provide this file if you wanted to override the default file extension to converter mappings provided.

Examples:

```
/E="Custom Extension To Converter Map"
```

/W - Wait Time

Use this switch wait to the specified number of seconds for the Document Conversion Service to be running and available to convert documents. If Document Conversion Service is already running the command executes immediately. If the Document Conversion Service is not running in the timeout period specified, the command will return with an error.

If this argument is not specified the command will return immediately with an error if Document Conversion Service is not running.

Example:

```
/W=300
```

/SIL - Smart Inspect Logging File

Smart Inspect Log files are a tracing of the entire conversion process and are not the same as the conversion results log files created when a conversion fails. These logs can be viewed using the SmartInspect Redistributable Console included with Document Conversion Service.

These log files are automatically deleted when conversion succeeds. To keep the log files on success use the custom setting *AlwaysKeepProcessingLoggingFiles* as shown below.

The default location for this file is the TEMP folder. Each logging file is assigned a unique date, time and thread prefix followed by "_PNConvertFileList.sil", such as *2014_09_11_2_38_00_PM_4_PNConvertFileList.sil*.

Use this argument to specify a custom path and optional file name for the SmartInspect logging file (*.sil) created by this utility. The /SIL switch can take a folder, or a path to a filename. If a path without a trailing backslash is provided, the last part of the path is assumed to be a filename.

Note: The double ending backslash used when specifying a folder for the /SIL switch is required for the command line path to be parsed correctly.

/SIL=	Is interpreted as...
"C:\Test\LogFile"	Create the SmartInspect log file as C:\Test\LogFile.sil.
"C:\Test\LogFile\"	Create the SmartInspect log file as C:\Test\LogFile\datetime_PNConvertFileList.sil
"C:\Test\LogFile\ConvertFileCustom.sil"	Create the SmartInspect log file as C:\Test\LogFile\ConvertFileCustom.sil

The following settings can be used to control the creation and naming of the logging file. These settings are all passed using the /D switch.

Custom Setting	Description
RemoveDateTimePrefixOnProcessingLoggingFiles	Pass <i>True</i> to disable the adding of the unique date, time and thread prefix when a custom file name has not been specified in the <i>ConvertFileProcessLoggingPath</i> parameter.
KeepFailedProcessingLoggingFiles	Pass as <i>False</i> to disable the automatic creation of SmartInspect logging files when conversion fails. This setting can be overridden by <i>AlwaysKeepProcessingLoggingFiles</i> .
AlwaysKeepProcessingLoggingFiles	When set to <i>True</i> , the SmartInspect logging files are always created in the %TEMP% or other specified folder for both successful and failed conversions. If set to <i>False</i> , no logging files are created. This setting will override the <i>KeepFailedProcessingLoggingFiles</i> setting.

Examples:

Pass a custom folder and remove the prefix, each run will overwrite the log file C:\PEERNET\Logs\PNConvertFileList.sil.

```
/SIL="C:\PEERNET\Logs\" /D="RemoveDateTimePrefixOnProcessingLoggingFiles:TRUE"
```

Pass a custom folder and log file name and remove the prefix. Each run will overwrite the logging file C:\PEERNET\Logs\MyLogFile.sil.

```
/SIL="C:\PEERNET\Logs\MyLogFile" /D="RemoveDateTimePrefixOnProcessingLoggingFiles:TRUE"
```

Don't save any SmartInspect log files at all.

```
/D="AlwaysKeepProcessingLoggingFiles:FALSE"
```

/I - Input text file path

The collection of files to be converted can be passed as a text file containing a list of files, one each per line. Optionally you can specify individual save locations for each file by listing the file and directory, separated by a semi-colon(;) on each line. The full path or a UNC path to the source file and optional directory must be given for the files listed in the input text file and as command line arguments; relative paths are not supported.

The input text file should follow the following format:

```
C:\Input\WordFiles\File1.doc
C:\Input\WordFiles\File2.docx;C:\OutputPath\WordFiles\
C:\Input\PDF\File3.pdf;C:\OutputPath\PDFFiles\
\\server\share\Input\scans\scan1.tif
```

/C - Convert on a Remote Computer (DCOM)

If Document Conversion Service is running on a different computer, use this switch to pass the name of the remote computer and the path of a shared location that both computers have access to. Separate the name of the remote computer and the path to the shared folder location with a semi-colon.

When converting remotely, the client redistributable, PNDocConvClientSetup_3.0.exe, must be installed on the computer running this command line utility. The client setup install program is included as part of the Document Conversion Service install and can be found in the **\Samples\Redist** folder in your product installation folder.

Examples:

```
/C="DOCCONV_SERVER;\\DOCCONV_SERVER\DCSREMOTE"
```

/T - Alternate Temp Folder

This is an advanced setting that should not be needed in most cases. When converting files, the conversion tool copies each file and performs the conversion in temporary *staging* and *working* folders created on demand in the default Windows temp folder. When dealing with long path and file names the default folders created can occasionally cause path names that are too long to process. When this happens this switch can be used to set the temporary folder to a shorter path to allow processing.

This setting is overridden if the /C option for remote conversion is being used with its own path to a shared location for conversion.

Examples:

```
/T="C:\PNTemp\\"
```

/? - Display Help

When passed as the only argument this switch will display help for this command.

file[;save location]

The full path to the file to convert. You can list more than one on the command line. Like the input text file, you can pass in a semi-colon(;) separated file-directory pair here as well.

- If the path to the file includes spaces it must be enclosed in quotes.
- If the file doesn't exist, the conversion will fail.

DCSConvertFolder

A command line utility to walk a folder and convert all files, or all files matching a search filter, using Document Conversion Service. The utility can optionally also process all subfolders under the starting folder as well. The Document Conversion Service must be running, either locally or on a remote computer for the files to be converted. If it is not running the command will return immediately with an error.

Any sorting options applied only control the order in which the files are picked up from the directory. Sorting does not guarantee the order the files are processed in, only that files sorted to the top of the list are submitted for conversion first. A smaller file further down the list might finish before a larger file that was first in the list.

```
DCSConvertFolder /P=profile [/R] [/F=filter] [/X=exclude filter] [/S=save location]
[/O] [/NE] [/L] [/D="name:value"] [/E=extension map]
[/FAIL=failed results log file location] [/SIL=conversion log file path]
[/W=wait time] [/C=remote computer name;remote scratch folder]
[/T=alternate temp folder]
[/NAME=sort files by name] [/CREATED=sort files by creation date]
[/MODIFIED=sort files by name] [/DESC=sort files in descending order (Z-A, 9-0)]
folder
```

Sample Command Lines

Convert all files in a folder to TIFF images:

```
DCSConvertFolder /P="TIFF 200dpi Monochrome" "C:\Test\Input"
```

Sends all files in the folder C:\Test\Input to Document Conversion Service to be converted using the settings contained in the conversion profile *TIFF 200dpi Monochrome.xml*. Any folders under C:\Test\Input are not processed.

Upon successful conversion each output file is placed in a folder named *.converted* created under the C:\Test\Input folder. Each output file is named using the base name and file extension of the original file, plus the extension of the file type you are creating.

If a file of that name already exists in the *.converted* folder the conversion will fail and a *.failed* folder will be created under the C:\Test\Input folder. A results log file, ending with *.failed.dcsresults*, is created for each failed file and saved to a new subfolder under the *.failed* folder. The subfolder is named using the date and time of the conversion to keep subsequent runs separate.

Convert all files in a folder to TIFF images, wait up to 5 minutes for the conversion service to start:

```
DCSConvertFolder /P="TIFF 200dpi Monochrome" /W=300 "C:\Test\Input"
```

Sends all files in the folder C:\Test\Input to Document Conversion Service to be converted using the settings contained in the conversion profile *TIFF 200dpi Monochrome.xml*. Any folders under C:\Test\Input are not processed.

If Document Conversion Service is not running, wait up to 5 minutes (300 seconds) for the conversion service to be available.

Convert all files in a folder, including subfolders, to TIFF images in a specific location:

```
DCSConvertFolder /R /P="TIFF 300dpi OptimizedColor" /S="C:\Test\Output"
"C:\Test\Input"
```

Walks the folder C:\Test\Input and any folders underneath and sends all the files found to Document Conversion Service to be converted using the settings contained in the conversion profile *TIFF 300dpi OptimizedColor.xml*.

Upon successful conversion each output file is placed in the C:\Test\Output folder in a directory structure that mirrors the source folder directory structure.

If a file does not convert, a subfolder named *.failed* is created in the same location as the input file. A results log named by appending *.failed.dcsresults* to the input file name is created and saved to a new subfolder under the *.failed* folder. The new subfolder is named using the date and time of the conversion to keep subsequent runs separate.

To store all of the failed file information in a separate location, see the /FAIL option.

Convert all Word and Excel files in the folder, including subfolders, to vector PDF documents in a specific location:

```
DCSConvertFolder /R /F="*.doc|*.docx|*.xls|*.xlsx" /X="12-01*" /S="C:\Test\Output\"
/P="C:\Test\Adobe PDF Multipage.xml" "C:\Test\Input"
```

Walks the folder C:\Test\Input and any folders underneath and sends all Word files ending in .doc and .docx and all Excel files ending in .xls and .xlsx to Document Conversion Service to be converted to vector PDF using the settings contained in the conversion profile *Adobe PDF Multipage.xml*. Any files or folders that begin with "12-01" are excluded.

Upon successful conversion each output file is placed under the C:\Test\Output folder in a directory structure that mirrors the source folder directory structure.

Failed conversion results logs are saved in a *.failed* folder created in the same location as the source file. A results log named by appending *.failed.dcsresults* to the input file name is created and saved to a new subfolder under the *.failed* folder. The new subfolder is named using the date and time of the conversion to keep subsequent runs separate.

Convert a folder of documents to vector PDF, overwrite any existing files, and save the results log:

```
DCSConvertFolder /P="Adobe PDF Multipage" /O /L "C:\Test\Input"
```

Sends all files in the folder C:\Test\Input to Document Conversion Service to be converted to vector PDF using the settings contained in the conversion profile *Adobe PDF Multipage.xml*. Any folders under C:\Test\Input are not processed.

Upon successful conversion each output file is placed in a folder named *.converted* created under the C:\Test\Input folder. Any files of the same name that already exist in that folder are overwritten.

A conversion results log file for each file converted will be also be saved in the *.converted* folder. The name of the results log file is based on the name of the original source file appended with *.succeeded.dcsresults*.

Failed conversion results logs are saved in a *.failed* folder created in the same location as the source file. A results log named by appending *.failed.dcsresults* to the input file name is created and saved to a new subfolder under the *.failed* folder. The new subfolder is named using the date and time of the conversion to keep subsequent runs separate.

Convert a folder of documents to vector PDF, strip off the source extension and save the output and results log files to a specific location:

```
DSCSConvertFolder /R /P="Adobe PDF Multipage" /NE /S="C:\Test\Output"  
/L /FAIL="C:\Test\FailedResults\\" "C:\Test\Input"
```

Walks the folder C:\Test\Input and any folders underneath and creates vector PDF files from all documents found. The type of PDF created is controlled by the settings in the conversion profile *Adobe PDF Multipage.xml*.

Upon successful conversion each output file is placed under the C:\Test\Output folder in a directory structure that mirrors the source folder directory structure. The /NE flag causes the output file to be named using the base name of the original file, plus the extension of the file type you are creating. If a file of that name already exists in the folder the conversion will fail.

A conversion results log file for each file will be also be saved in the C:\Test\Output folder in the same mirrored directory structure. The name of the results log file is based on the original source file and appended with *.succeeded.dcsresults* to indicate its conversion status.

If the conversion did not succeed, the results log is named by appending *.failed.dcsresults* to the input file name and placing this file into a subfolder named with the current date and time created under the specified folder C:\Test\FailedResults.

Note: The double ending backslash used when specifying the folder for the /FAIL switch is required for the command line path to be parsed correctly.

Use the command line argument *D="UseDateTimeInFailedFolder:FALSE"* to store the results log file directly in the folder C:\FailedResults.

Controlling the Number of Documents Processed in Parallel

When converting a folder of files, the number of documents that can be passed in parallel (*at the same time*) to Document Conversion Service to be converted is automatically determined based on the number of CPU's and cores on your system multiplied by 1.5. We recommend that you allow this value to be determined automatically, but if needed, you can specify exactly how many documents you want to process in parallel by adding the following line into the profile you are using.

Please note that this value is completely separate from the value of the same name used by the Document Conversion Service configuration. Also, keep in mind that setting this to a value that is too high for the capabilities of the computer can cause the computer to work very slowly.

```
<add Name="NumberOfDocumentsInParallel" Value="10"/>
```

You can also pass this directly on the command line using the /D option.

```
DSCSConvertFolder /P="TIFF 200dpi Monochrome.xml" /D="NumberOfDocumentsInParallel:6"  
"C:\Test\Input"
```

Command Line Arguments

Command line switches are not case-sensitive and can be entered in either upper or lower case.

/R - Include Subfolders (Recurse)

Use this switch to also search any subfolders under the source folder when building the list of files to be passed to Document Conversion Service to be converted.

/F - File Filter

A filter can be provided using this switch to only process certain types of files. Multiple file filters can be combined using the pipe (|) character. Hidden and system files are ignored, and the search pattern filters files based on a regular expression match of the long name of a file.

When this switch is not specified all files in the folder are (*.*) passed to Document Conversion Service to be processed.

Examples:

Convert PDF only: /F="*.pdf"

Convert Word, Excel and PDF only: /F="*.doc|*.docx|*.xls|*.xlsx|*.pdf"

Convert all Word files starting with MEMO: /F="MEMO*.doc"

/X - Exclude File Filter

A exclude file filter can be provided to take the returned file list gathered using the /F file filter and exclude any files that match a pattern. Multiple patterns can be combined using the pipe (|) character. By default no files are excluded.

Examples:

Exclude Word and Excel 2010 documents: /X="*.docx|*.xlsx"

Exclude all files starting with "Draft": /X="Draft*.*"

/S - The Save Location

Pass in the full path to the folder in which to save the new files. If the /R switch is used the original directory structure is maintained.

- If the path includes spaces it must be enclosed in quotes.
- If the path is specified but doesn't exist, the conversion will fail.
- If a file of the same name already exists in the save file location, the conversion will fail. The /O option can be used to enable file overwriting, which is off by default.

If this argument is not specified, a *.converted* folder is created in the same location as each source file and all output files are saved there. On subsequent processing of the same folder with the /R switch enabled, any *.converted* folders are ignored.

Example:

/S="C:\Converted Files\Test"

/O - Overwrite Always

Enables overwrite mode so that existing files of the same name are overwritten with the new file. If the overwrite switch is not specified, the conversion of that file in the list of files will fail if a file of the same name already exists in the output folder.

/NE - No Extension

Specify this option if you do not want the original file name extension as part of your output file name. Normally the name of the each output file is created using the base name and file extension of the original file to prevent name collision when you have two files in the folder with the same base name.

/L - Results Log

The results log file is an XML file containing a complete snapshot of the conversion information for each file converted. Normally only saved for failed conversions, the /L argument enables creation of the results log file when the conversion succeeds. The results log file is placed in the same location as the converted files.

The name of the results log file is based on the name of the original file and also indicates the conversion status. For example, when converting *Document.doc*, a successful conversion will create a log file named *Document.doc.succeeded.dcsresults*, while a failed conversion would be named *Document.doc.failed.dcsresults*.

The results log file for a successful conversion is always copied to the output location with the converted files when this flag is used.

In the case of a failed conversion, the log file is always created. See the /FAIL switch to control the location and creation of the failed results log files.

The result log files can later be passed to the [DCSExtractResults](#) command line utility to extract information such as all files created or any errors encountered during conversion. You can search a folder for the results log files using the [DCSCreateFileList](#) utility.

/FAIL - Failed Results Log File Location

In the case of a failed conversion, the conversion results log file is always created. The default behavior is to create a *.failed* folder in the same location as the source file and save the conversion results log file to a new subfolder under the *.failed* folder. The subfolder is named using the date and time of the conversion to keep subsequent runs separate.

This argument allows you to override the default use of the *.failed* folder and to provide a specific folder in which to store the failed results log files. The name of the results log file is based on the name of the original file and also indicates the conversion status. For example, when converting *Document.doc*, a failed conversion would be named *Document.doc.failed.dcsresults*.

You can suppress the use of the date and time subfolder by passing the *UseDateTimeInFailedFolder* setting using the /D switch.

Note: The double ending backslash used when specifying the folder for the /FAIL switch is required for the command line path to be parsed correctly.

Examples:

```
/FAIL="C:\ConvertedFiles\Failed\\" /D="UseDateTimeInFailedFolder:FALSE"
```

If you do not want to create the failed results log files at all, you can use the /D switch to pass the *KeepFailedItemResultsFiles* setting as *false*.

On the command line:

```
/D="KeepFailedItemResultsFiles:False"
```

In a conversion profile:

```
<add Name="KeepFailedItemResultsFiles" Value="False"/>
```

The result log files can later be passed to the [DCSExtractResults](#) command line utility to extract information such as the source file used or any errors encountered during conversion. You can search a folder for the results log files using the [DCSCreateFileList](#) utility.

/P - Conversion Profile

This is a required argument. The type of file created is controlled by supplying a conversion profile using this switch. The profiles are referenced by passing in the name of the profile XML file, with or without the XML extension. See [Creating and Customizing Profiles](#) for more information about the contents of the profiles, a list of profiles included with Document Conversion Service, and how to create your own.

Examples:

```
/P="TIFF 300dpi Color Fax"  
/P="TIFF 204x196dpi Monochrome Fax.xml"
```

/D - Define Setting

Individual profile settings can be supplied on the command line using this switch. This switch can be specified multiple times for separate settings and any settings passed here will override the settings in the profile.

Any name-value pair that can be written in a profile can be passed through this parameter. This includes options to control the conversion settings as well as the behavior of the individual converters as well. See [Creating and Customizing Profiles](#) for more information about the name-value pairs that can be used.

Examples:

These first two are settings that control the converter options, such as what pages to print, and the output that PowerPoint will print.

```
/D="PrintRange:1-5"  
/D="PowerPoint.PrintOptionsOutputType:PrintOutputNotesPages"
```

These two settings control the output file creation options, and would override or add to the settings in the conversion profile passed using the /P switch.

```
/D="Image Options;Fax Resolution:3"  
/D="TIFF File Format;BW compression:Group3-2D"
```

These two settings control where the failed results log files are created and are most often used along with the /FAIL switch to control where the results log files are saved.

```
/D="KeepFailedItemResultsFiles:TRUE"  
/D="UseDateTimeInFailedFolder:FALSE"
```

/E - File Extension Mapping

A file extension mapping profile uses the extension of the source file to determine what converter will be used to convert the file. Like the conversion profiles, this file is also an XML file. This switch is optional and an internal default mapping is provided. You would only need to provide this file if you wanted to override the default file extension to converter mappings provided.

Examples:

```
/E="Custom Extension To Converter Map"
```

/W - Wait Time

Use this switch wait to the specified number of seconds for the Document Conversion Service to be running and available to convert documents. If Document Conversion Service is already running the command executes immediately. If the Document Conversion Service is not running in the timeout period specified, the command will return with an error.

If this argument is not specified the command will return immediately with an error if Document Conversion Service is not running.

Example:

```
/W=300
```

/C - Convert on a Remote Computer (DCOM)

If Document Conversion Service is running on a different computer, use this switch to pass the name of the remote computer and the path of a shared location that both computers have access to. Separate the name of the remote computer and the path to the shared folder location with a semi-colon.

When converting remotely, the client redistributable, PNDocConvClientSetup_3.0.exe, must be installed on the computer running this command line utility. The client setup install program is included as part of the Document Conversion Service install and can be found in the **\Samples\Redist** folder in your product installation folder.

Examples:

```
/C="DOC CONV_SERVER;\\DOC CONV_SERVER\DCSREMOTE"
```

/SIL - Smart Inspect Logging File

Smart Inspect Log files are a tracing of the entire conversion process and are not the same as the conversion results log files created when a conversion fails. These logs can be viewed using the SmartInspect Redistributable Console included with Document Conversion Service.

These log files are automatically deleted when conversion succeeds. To keep the log files on success use the custom setting *AlwaysKeepProcessingLoggingFiles* as shown below.

The default location for this file is the TEMP folder. Each logging file is assigned a unique date, time and thread prefix followed by "_PNConvertFolder.sil", such as *2014_09_11_2_38_00_PM_4_PNConvertFolder.sil*.

Use this argument to specify a custom path and optional file name for the SmartInspect logging file (*.sil) created by this utility. The /SIL switch can take a folder, or a path to a filename. If a path without a trailing backslash is provided, the last part of the path is assumed to be a filename.

Note: The double ending backslash used when specifying a folder for the /SIL switch is required for the command line path to be parsed correctly.

/SIL=	Is interpreted as...
"C:\Test\LogFile"	Create the SmartInspect log file as C:\Test\LogFile.sil.
"C:\Test\LogFile\"	Create the SmartInspect log file as C:\Test\LogFile\datetime_PNConvertFolder.sil
"C:\Test\LogFile\ConvertFileCustom.sil"	Create the SmartInspect log file as C:\Test\LogFile\ConvertFileCustom.sil

The following settings can be used to control the creation and naming of the logging file. These settings are all passed using the /D switch.

Custom Setting	Description
RemoveDateTimePrefixOnProcessingLoggingFiles	Pass <i>True</i> to disable the adding of the unique date, time and thread prefix when a custom file name has not been specified in the <i>ConvertFileProcessLoggingPath</i> parameter.
KeepFailedProcessingLoggingFiles	Pass as <i>False</i> to disable the automatic creation of SmartInspect logging files when conversion fails. This setting can be overridden by <i>AlwaysKeepProcessingLoggingFiles</i> .
AlwaysKeepProcessingLoggingFiles	When set to <i>True</i> , the SmartInspect logging files are always created in the %TEMP% or other specified folder for both successful and failed conversions. If set to <i>False</i> , no logging files are created. This setting will override the <i>KeepFailedProcessingLoggingFiles</i> setting.

Examples:

Pass a custom folder and remove the prefix, each run will overwrite the log file C:\PEERNET\Logs\PNConvertFolder.sil.

```
/SIL="C:\PEERNET\Logs\" /D="RemoveDateTimePrefixOnProcessingLoggingFiles:TRUE"
```

Pass a custom folder and log file name and remove the prefix. Each run will overwrite the logging file C:\PEERNET\Logs\MyLogFile.sil.

```
/SIL="C:\PEERNET\Logs\MyLogFile" /D="RemoveDateTimePrefixOnProcessingLoggingFiles:TRUE"
```

Don't save any SmartInspect log files at all.

```
/D="AlwaysKeepProcessingLoggingFiles:FALSE"
```


/T - Alternate Temp Folder

This is an advanced setting that should not be needed in most cases. When converting files, the conversion tool copies each file and performs the conversion in temporary *staging* and *working* folders created on demand in the default Windows temp folder. When dealing with long path and file names the default folders created can occasionally cause path names that are too long to process. When this happens this switch can be used to set the temporary folder to a shorter path to allow processing.

This setting is overridden if the /C option for remote conversion is being used with its own path to a shared location for conversion.

Examples:

```
/T= "C:\PNTemp\\"
```

/NAME - Sort Files by Name

The file list picked up from the folder is sorted by file name in ascending order (0-9, A-Z).

/CREATED - Sort Files by Creation Date

The file list picked up from the folder is sorted by the files creation date file name in ascending order (0-9, A-Z).

/MODIFIED - Sort Files by Modified Date

The file list picked up from the folder is sorted by the files last modified date file name in ascending order (0-9, A-Z).

/DESC - Sort Files in Descending Order

Use with /NAME, /CREATED or /MODIFIED. Sorts the descending order (Z-A, 9-0).

/? - Display Help

When passed as the only argument this switch will display help for this command.

Folder

The folder containing the files to convert.

- If the path to the folder includes spaces it must be enclosed in quotes.
- If the folder doesn't exist, the conversion will fail.

DCSCombineFiles

A command line utility that accepts a text file containing a list of files, and/or a list of files provided on the command line, and combines all of the files together into a single file or a collection of serialized pages using Document Conversion Service. The files are appended together in the order in which they are received.

The Document Conversion Service must be running, either locally or on a remote computer for the files to be combined. If it is not running the command will return immediately with an error.

```
DCSCombineFiles /P=profile /S=save location /N=output name
                [/O] [/L] [/E=extension map]
                [/C=remote computer name;remote scratch folder]
                [/FAIL=failed results log file location] [/SIL=conversion log file path]
                [/D="name:value"] [/W=wait time]
                [/I=input text file path]
                [/T=alternate temp folder]
                "file" "file" ...
```

Sample Command Lines

Combine all files on command line into a single TIFF image:

```
DCSCombineFiles /S="C:\Test\Output" /N="CombinedFiles" /P="TIFF 200dpi Monochrome"
                "C:\Input\File1.doc" "C:\Input2\File2.doc"
```

Sends the files C:\Input\File1.doc and C:\Input2\File2.doc to Document Conversion Service to be converted using the settings contained in the conversion profile *TIFF 200dpi Monochrome*.

The converted files are saved as a single TIFF image, named *CombinedFiles.tif* in the output folder C:\Test\Output. The files are combined in the order in which they are returned by the underlying file system.

If an output file with the same name already exists, or if one of the files in the combine set fails to convert, the combine will fail and a results log file will be placed in a folder named *.failed* created in the save location. The results log file name will be *"PNCombineFiles_"*, followed by a date and time stamp and ending with *failed.dcsresults*. This can be controlled with the */FAIL* switch.

To overwrite an existing file the */O* switch would need to be added to the above command.

Convert all files in the input file to a multipage PDF:

```
DCSCombineFiles /S="C:\Test\Output" /N="CombinedFiles"
                /P="PDF 300dpi OptimizedColor"
                /I="C:\Test\Files.txt"
```

Sends the files listed in the text file C:\Test\Files.txt to Document Conversion Service to be converted using the settings contained in the conversion profile *PDF 300dpi OptimizedColor*.

The converted files are saved as a multipaged PDF file named *CombinedFiles.pdf* in the output folder C:\Test\Output. The files are combined in the order they are listed in the input file.

Upon successful conversion each output file is placed under the C:\Test\Output folder.

If a file with the same name already exists, or if one of the files in the combine set fails to convert, the conversion will fail and a results log file will be placed in a folder named *.failed* created in the save location. Where this file is saved can be controlled with the */FAIL* switch.

The results log file name starts with *PNCombineFiles*, contains a date and time stamp and ends with *.failed.dcsresults*.

Convert all files in the input file and command line to a multipage PDF in that order:

```
DCSCombineFiles /S="C:\Test\Output" /N="CombinedFiles"
                /P="PDF 300dpi OptimizedColor" /I="C:\Test\Files.txt"
                "C:\Test\EndOfCombine.doc"
```

Sends the files listed in the text file "C:\Test\Files.txt", followed by the file "C:\Test\EndOfCombine.doc" specified on the command line, in that order, to Document Conversion Service to be converted using the settings contained in the conversion profile *PDF 300dpi OptimizedColor*.

The converted files are saved as a multipaged PDF file named *CombinedFiles.pdf* in the output folder C:\Test\Output. The files are combined in the order they are listed in the input file.

Upon successful conversion each output file is placed under the C:\Test\Output folder.

If a file with the same name already exists, or if one of the files in the combine set fails to convert, the combine will fail and a results log file will be placed in a folder named *.failed* created in the save location. Where this file is saved can be controlled with the */FAIL* switch.

The results log file name starts with *PNCombineFiles*, contains a date and time stamp and ends with *.failed.dcsresults*.

Combine all files on command line into a single TIFF image, save conversion results logs to a specific location:

```
DCSCombineFiles /L /S="C:\Test\Output" /N="CombinedFiles"
                /P="TIFF 200dpi Monochrome" /FAIL="C:\Test\Output\Failed\\"
                "C:\Input\File1.doc" "C:\Input2\File2.doc"
```

Sends the files C:\Input\File1.doc and C:\Input2\File2.doc to Document Conversion Service to be converted using the settings contained in the conversion profile *TIFF 200dpi Monochrome*.

The converted files are saved as single TIFF image, named *CombinedFiles.tif* in the output folder C:\Test\Output. The files are combined in the order given on the command line - File1.doc followed by File2.doc.

If a file with the same name already exists, or if one of the files in the combine set fails to convert, the combine will fail and a conversion results log file will be placed into the folder C:\Test\Output\Failed.

The conversion results log file name starts with *PNCombineFiles*, contains a date and time stamp and ends with *.failed.dcsresults*. You can use the */D* parameter *UseDateTimeInFailedFolder* to remove the date and time stamp from the results log file name.

To overwrite an existing file the */O* switch would need to be added to the above command.

Command Line Arguments

Command line switches are not case-sensitive and can be entered in either upper or lower case.

/S - The Save Location

This is a required argument. Pass in the full path to the folder in which to save the new files.

- If the path includes spaces it must be enclosed in quotes.
- If the path doesn't exist, the conversion will fail.
- If a file of the same name already exists in the save file location, the conversion will fail. To enable file overwriting, use the /O option.

Example:

```
/S="C:\Converted Files\Test"
```

/N - Output File Name

This is a required argument and specifies the name to use for the output file. The default file extension for the type of file being created will always be added to the name provided here.

Example:

```
/N="CombinedOutput_06_15_2012"
```

/O - Overwrite Always

Enables overwrite mode so that existing files of the same name are overwritten with the new file. If overwrite is not specified the combine action will fail if a file of the same name already exists in the save location.

/L - Results Log

The results log file is an XML file containing a complete snapshot of the combine request. Normally only saved for failed conversions, the /L argument enables creation of the results log file when the conversion succeeds.

All results log files for this command line utility start with *PNCombineFiles_*, contain a date and time stamp and end with the conversion status.

When the combine has succeeded, the results log file is placed in the same folder as the output (specified using the /S switch) and would have a name similar to the following:

```
PNCombineFiles_2013_05_31_2_50_05_PM_3.succeeded.dcsresults
```

The bold text in the name will change for each file and is based on the date and time of the run and an internal counter. You can suppress the use of the date and time information in the file name by passing *false* for the *UseDateTimeInFailedFolder* setting using the /D switch.

In the case of a failed conversion, the log file is always created. See the /FAIL switch to control the location and creation of the failed results log files.

The result log files can later be passed to the [DCSExtractResults](#) command line utility to extract information such as all files created or any errors encountered during conversion. You can search a folder for the results log files using the [DCSCreateFileList](#) utility.

/FAIL - Combine Results Log File Location

In the case of a failed combine, the combine results log file is always created. When the combine does not succeed, a *.failed* folder is created in the save folder location (provided by the /S switch) and the results log files are stored there.

The name of the results log when the combine does not succeed will be similar to the following:

```
PNCombineFiles_2013_05_31_2_50_05_PM_3.failed.dcsresults
```

The bold text in the name will change each time a combine command is run and is based on the date and time of the run and an internal counter.

This argument allows you to override the default use of the *.failed* folder and to provide a specific folder in which to store the failed results log file. You can suppress the use of the date and time information in the file name by passing *false* for the *UseDateTimeInFailedFolder* setting using the /D switch.

Note: The double ending backslash used when specifying the folder for the /FAIL switch is required for the command line path to be parsed correctly.

Examples:

```
/FAIL="C:\ConvertedFiles\Failed\\" /D="UseDateTimeInFailedFolder:FALSE"
```

If you do not want to create the failed results log files at all, you can use the /D switch to pass the *KeepFailedItemResultsFiles* setting as *false*.

On the command line:

```
/D="KeepFailedItemResultsFiles:False"
```

In a conversion profile:

```
<add Name="KeepFailedItemResultsFiles" Value="False"/>
```

The result log files can later be passed to the [DCSExtractResults](#) command line utility to extract information such the source file used or any errors encountered during conversion. You can search a folder for the results log files using the [DCSCreateFileList](#) utility.

/P - Conversion Profile

This argument is required. The type of file created is controlled by supplying a conversion profile using this switch. The profiles are referenced by passing in the name of the profile XML file, with or without the XML extension. See [Creating and Customizing Profiles](#) for more information about the contents of the profiles, a list of profiles included with Document Conversion Service, and how to create your own.

Examples:

```
/P="TIFF 300dpi Color Fax"  
/P="TIFF 204x196dpi Monochrome Fax.xml"
```

/D - Define Setting

Individual profile settings can be supplied on the command line using this switch. This switch can be specified multiple times for separate settings and any settings passed here will override the settings in the profile.

Any name-value pair that can be written in a profile can be passed through this parameter. This includes options to control the conversion settings as well as the behavior of the individual converters as well. See [Creating and Customizing Profiles](#) for more information about the name-value pairs that can be used.

Examples:

These first two are settings that control the converter options, such as what pages to print, and the output that PowerPoint will print.

```
/D="PrintRange:1-5"
```

```
/D="PowerPoint.PrintOptionsOutputType:PrintOutputNotesPages"
```

These two settings control the output file creation options, and would override or add to the settings in the conversion profile passed using the /P switch.

```
/D="Image Options;Fax Resolution:3"
```

```
/D="TIFF File Format;BW compression:Group3-2D"
```

These two settings control the where the failed results log files are created and are most often used along with the /FAIL switch to control where the results log files are saved.

```
/D="KeepFailedItemResultsFiles:TRUE"
```

```
/D="UseDateTimeInFailedFolder:FALSE"
```

/E - File Extension Mapping

A file extension mapping profile uses the extension of the source file to determine what converter will be used to convert the file before combining them together. Like the conversion profiles, this file is also an XML file. This switch is optional and an internal default mapping is provided. You would only need to provide this file if you wanted to override the default file extension to converter mappings provided.

Examples:

```
/E="Custom Extension To Converter Map"
```

/W - Wait Time

Use this switch wait to the specified number of seconds for the Document Conversion Service to be running and available to convert and combine documents. If Document Conversion Service is already running the command executes immediately. If the Document Conversion Service is not running in the timeout period specified, the command will return with an error.

If this argument is not specified the command will return immediately with an error if Document Conversion Service is not running.

Example:

```
/W=300
```

/I - Input text file path

The collection of files to be combined can be passed as a text file containing a list of files, one each per line. The full path or a UNC path to the source file must be given for the files listed in the input text file; relative paths are not supported.

- If the path to the file includes spaces it must be enclosed in quotes.
- If the file doesn't exist, the conversion will fail.

The files are combined together in the order in which they are listed in the folder. Any files were specified directly on the command line before this switch are combined before adding the files in the input text file. Any source files specified on the command line after this switch are combined after the files in the input text file.

The input text file should follow the following format:

```
C:\Input\WordFiles\File1.doc
C:\Input\WordFiles\File2.docx;C:\OutputPath\WordFiles\
C:\Input\PDF\File3.pdf;C:\OutputPath\PDFFiles\
\\server\share\Input\scans\scan1.tif
```

/C - Convert on a Remote Computer (DCOM)

If Document Conversion Service is running on a different computer, use this switch to pass the name of the remote computer and the path of a shared location that both computers have access to. Separate the name of the remote computer and the path to the shared folder location with a semi-colon.

When combining remotely, the client redistributable, PNDocConvClientSetup_3.0.exe, must be installed on the computer running this command line utility. The client setup install program is included as part of the Document Conversion Service install and can be found in the **\Samples\Redist** folder in your product installation folder.

Examples:

```
/C= "DOCCONV_SERVER; \\DOCCONV_SERVER\DCSREMOTE "
```

/SIL - Smart Inspect Logging File

Smart Inspect Log files are a tracing of the entire conversion process and are not the same as the conversion results log files created when a conversion fails. These logs can be viewed using the SmartInspect Redistributable Console included with Document Conversion Service.

These log files are automatically deleted when conversion succeeds. To keep the log files on success use the custom setting *AlwaysKeepProcessingLoggingFiles* as shown below.

The default location for this file is the TEMP folder. Each logging file is assigned a unique date, time and thread prefix followed by "_PNCombineFiles.sil", such as *2014_09_11_2_38_00_PM_4_PNCombineFiles.sil*.

Use this argument to specify a custom path and optional file name for the SmartInspect logging file (*.sil) created by this utility. The /SIL switch can take a folder, or a path to a filename. If a path without a trailing backslash is provided, the last part of the path is assumed to be a filename.

Note: The double ending backslash used when specifying a folder for the /SIL switch is required for the command line path to be parsed correctly.

/SIL=	Is interpreted as...
"C:\Test\LogFile"	Create the SmartInspect log file as C:\Test\LogFile.sil.
"C:\Test\LogFile\"	Create the SmartInspect log file as C:\Test\LogFile\datetime_PNCombineFiles.sil
"C:\Test\LogFile\ConvertFileCustom.sil"	Create the SmartInspect log file as C:\Test\LogFile\ConvertFileCustom.sil

The following settings can be used to control the creation and naming of the logging file. These settings are all passed using the /D switch.

Custom Setting	Description
RemoveDateTimePrefixOnProcessingLoggingFiles	Pass <i>True</i> to disable the adding of the unique date, time and thread prefix when a custom file name has not been specified in the <i>ConvertFileProcessLoggingPath</i> parameter.
KeepFailedProcessingLoggingFiles	Pass as <i>False</i> to disable the automatic creation of SmartInspect logging files when conversion fails. This setting can be overridden by <i>AlwaysKeepProcessingLoggingFiles</i> .
AlwaysKeepProcessingLoggingFiles	When set to <i>True</i> , the SmartInspect logging files are always created in the %TEMP% or other specified folder for both successful and failed conversions. If set to <i>False</i> , no logging files are created. This setting will override the <i>KeepFailedProcessingLoggingFiles</i> setting.

Examples:

Pass a custom folder and remove the prefix, each run will overwrite the log file C:\PEERNET\Logs\PNCombineFiles.sil.

```
/SIL="C:\PEERNET\Logs\" /D="RemoveDateTimePrefixOnProcessingLoggingFiles:TRUE"
```

Pass a custom folder and log file name and remove the prefix. Each run will overwrite the logging file C:\PEERNET\Logs\MyLogFile.sil.

```
/SIL="C:\PEERNET\Logs\MyLogFile" /D="RemoveDateTimePrefixOnProcessingLoggingFiles:TRUE"
```

Don't save any SmartInspect log files at all.

```
/D="AlwaysKeepProcessingLoggingFiles:FALSE"
```


/T - Alternate Temp Folder

This is an advanced setting that should not be needed in most cases. When converting files, the conversion tool copies each file and performs the conversion in temporary *staging* and *working* folders created on demand in the default Windows temp folder. When dealing with long path and file names the default folders created can occasionally cause path names that are too long to process. When this happens this switch can be used to set the temporary folder to a shorter path to allow processing.

This setting is overridden if the /C option for remote conversion is being used with its own path to a shared location for conversion.

Examples:

```
/T= "C:\PNTemp\\"
```

/? - Display Help

When passed as the only argument this switch will display help for this command.

File

The full path to the files to combine. You can list more than one on the command line.

The files are combined together in the order in which they listed on the command line. If any files were specified in an input text file using the /I switch before these files, the files listed in the input text file are combined before adding the files from the command line. If you specify the input text file after the files on the command line, the command line files are combined first, then the files listed in the input text file.

- If the path to the file includes spaces it must be enclosed in quotes.
- If the file doesn't exist, the conversion will fail.

DCSCombineFolder

A command line utility to walk a folder and combines all files, or all files matching a search filter, into a single file or a collection of serialized pages using Document Conversion Service. The utility can optionally also process all subfolders under the starting folder as well.

The order of the files in the combined file cannot be guaranteed and is dependent on the file system. Most cases they are alphabetical but can also be by creation time. Files from the root of the input folder are listed first, then all files from the subfolders when enabled. Subfolders are listed in alphabetical or creation time order, again dependent on the file system.

The Document Conversion Service must be running, either locally or on a remote computer for the files to be combined. If it is not running the command will return immediately with an error.

```
DCSCombineFolder /P=profile /S=save location /N=output name
                [/R] [/F=filter] [/X=exclude filter]
                [/O] [/L] [/E=extension map]
                [/C=remote computer name;remote scratch folder]
                [/FAIL=failed results log file location] [/SIL=conversion log file path]
                [/D="name:value"] [/W=wait time]
                [/T=alternate temp folder]
                [/NAME=sort files by name] [/CREATED=sort files by creation date]
                [/MODIFIED=sort files by name] [/DESC=sort files in descending order (Z-A, 9-0)]
                "folder"
```

Sample Command Lines

Combine all files in the folder into a single TIFF image:

```
DCSCombineFolder /S="C:\Test\Output" /N="CombinedFromFolder"
                /P="TIFF 200dpi Monochrome"
                "C:\Input\"
```

Searches for and send all files from the root of the C:\Input\ folder to Document Conversion Service to be converted using the settings contained in the conversion profile *TIFF 200dpi Monochrome*.

The converted files are combined and saved as single TIFF image, named *CombinedFromFolder.tif* in the output folder C:\Test\Output.

If the file C:\Test\Output\CombinedFromFolder.tif already exists, or if one of the files in the combine set fails to convert, the combine will fail and a results log file will be placed in a folder named *.failed* created in the save location. The results log file name will be *PNCombineFolder_*, followed by a date and time stamp and ending with *failed.dcsresults*. This can be controlled with the /FAIL switch.

To overwrite an existing file the /O switch would need to be added to the above command.

Convert all files in the folder to a multipage PDF:

```
DCSCombineFolder /S="C:\Test\Output" /N="CombinedFromFolder"
                /P="PDF 300dpi OptimizedColor"
                "C:\Input\"
```

Searches for and sends all files from the root of C:\Input\ folder to Document Conversion Service to be converted using the settings contained in the conversion profile *PDF 300dpi OptimizedColor*.

The converted files are combined and saved as a multipaged PDF file named *CombinedFromFolder.pdf* in the output folder C:\Test\Output.

If the file C:\Test\Output\CombinedFromFolder.pdf already exists, or if one of the files in the combine set fails to convert, the conversion will fail and a results log file will be placed in a folder named *.failed* created in the save location. Where this file is saved can be controlled with the /FAIL switch.

The results log file name starts with *PNCombineFolder*, contains a date and time stamp and ends with *.failed.dcsresults*.

Convert all Word documents in a folder and all subfolders to a multipage PDF:

```
DCSCombineFolder /S="C:\Test\Output" /N="CombinedFromFolder" /R /F="*.doc|*.docx"  
/P="PDF 300dpi OptimizedColor"  
"C:\Input\"
```

Searches C:\Input\ and any subfolders for Word documents to send to Document Conversion Service to be converted using the settings contained in the conversion profile *PDF 300dpi OptimizedColor*.

The converted files are combined and saved as a multipaged PDF file named *CombinedFromFolder.pdf* in the output folder *C:\Test\Output*.

If the file *C:\Test\Output\CombinedFromFolder.pdf* already exists, or if one of the files in the combine set fails to convert, the conversion will fail and a results log file will be placed in a folder named *.failed* created in the save location. Where this file is saved can be controlled with the */FAIL* switch.

The results log file name starts with *PNCombineFolder*, contains a date and time stamp and ends with *.failed.dcsresults*.

Combine all files except PDF to a new PDF file, save conversion results logs to a specific location:

```
DCSCombineFolder /R /L /S="C:\Test\Output" /N="CombinedFromFolder"  
/P="PDF 300dpi OptimizedColor" /X="*.pdf"  
/FAIL="C:\Test\Output\Failed\"  
"C:\Input\"
```

Searches C:\Input\ and any subfolders (/R) for all files except PDF documents (/X="*.pdf") to send to Document Conversion Service to be converted using the settings contained in the conversion profile *PDF 300dpi OptimizedColor*.

The converted files are saved as single PDF file, *CombinedFromFolder.pdf*, in the output folder *C:\Test\Output*.

If a file with the same name already exists, or if one of the files in the combine set fails to convert, the combine will fail and a conversion results log file will be placed into the folder *C:\Test\Output\Failed*.

The conversion results log file name starts with *PNCombineFolder*, contains a date and time stamp and ends with *failed.dcsresults*. You can use the */D* parameter *UseDateTimeInFailedFolder* to remove the date and time stamp from the results log file name.

To overwrite an existing file the */O* switch would need to be added to the above command.

Command Line Arguments

Command line switches are not case-sensitive and can be entered in either upper or lower case.

/R - Include Subfolders (Recurse)

Use this switch to also search any subfolders under the source folder when building the list of files to be passed to Document Conversion Service to be converted.

/F - File Filter

A filter can be provided using this switch to only process certain types of files. Multiple file filters can be combined using the pipe (|) character. Hidden and system files are ignored, and the search pattern filters files based on a regular expression match of the long name of a file.

When this switch is not specified all files in the folder are (*.*) passed to Document Conversion Service to be processed.

Examples:

Convert PDF only: /F="*.pdf"

Convert Word, Excel and PDF only: /F="*.doc|*.docx|*.xls|*.xlsx|*.pdf"

Convert all Word files starting with MEMO: /F="MEMO*.doc"

/X - Exclude File Filter

A exclude file filter can be provided to take the returned file list gathered using the /F file filter and exclude any files that match a pattern. Multiple patterns can be combined using the pipe (|) character. By default no files are excluded.

Examples:

Exclude Word and Excel 2010 documents: /X="*.docx|*.xlsx"

Exclude all files starting with "Draft": /X="Draft*.*"

/S - The Save Location

This is a required argument. Pass in the full path to the folder in which to save the new files.

- If the path includes spaces it must be enclosed in quotes.
- If the path doesn't exist, the conversion will fail.
- If a file of the same name already exists in the save file location, the conversion will fail. To enable file overwriting, use the /O option.

Example:

```
/S="C:\Converted Files\Test"
```

/N - Output File Name

This is a required argument and specifies the name to use for the output file. The default file extension for the type of file being created will always be added to the name provided here.

Example:

```
/N="CombinedOutput_06_15_2012"
```

/O - Overwrite Always

Enables overwrite mode so that existing files of the same name are overwritten with the new file. If overwrite is not specified the combine action will fail if a file of the same name already exists in the save location.

/L - Results Log

The results log file is an XML file containing a complete snapshot of the combine request. Normally only saved for failed conversions, the /L argument enables creation of the results log file when the conversion succeeds.

All results log files for this command line utility start with *PNCombineFolder_*, contain a date and time stamp and end with the conversion status.

When the combine has succeeded, the results log file is placed in the same folder as the output (specified using the /S switch) and would have a name similar to the following:

```
PNCombineFolder_2013_05_31_2_50_05_PM_3.succeeded.dcsresults
```

The bold text in the name will change for each file and is based on the date and time of the run and an internal counter. You can suppress the use of the date and time information in the file name by passing *false* for the *UseDateTimeInFailedFolder* setting using the /D switch.

In the case of a failed conversion, the log file is always created. See the /FAIL switch to control the location and creation of the failed results log files.

The result log files can later be passed to the [DCSExtractResults](#) command line utility to extract information such as all files created or any errors encountered during conversion. You can search a folder for the results log files using the [DCSCreateFileList](#) utility.

/FAIL - Combine Results Log File Location

In the case of a failed combine, the combine results log file is always created. When the combine does not succeed, a *.failed* folder is created in the save folder location (provided by the /S switch) and the results log files are stored there.

The name of the results log when the combine does not succeed will be similar to the following:

```
PNCombineFolder_2013_05_31_2_50_05_PM_3.failed.dcsresults
```

The bold text in the name will change each time a combine command is run and is based on the date and time of the run and an internal counter.

This argument allows you to override the default use of the *.failed* folder and to provide a specific folder in which to store the failed results log file. You can suppress the use of the date and time information in the file name by passing *false* for the *UseDateTimeInFailedFolder* setting using the /D switch.

Note: The double ending backslash used when specifying the folder for the /FAIL switch is required for the command line path to be parsed correctly.

Examples:

```
/FAIL="C:\ConvertedFiles\Failed\\" /D="UseDateTimeInFailedFolder:FALSE"
```

If you do not want to create the failed results log files at all, you can use the /D switch to pass the *KeepFailedItemResultsFiles* setting as *false*.

On the command line:

```
/D="KeepFailedItemResultsFiles:False"
```

In a conversion profile:

```
<add Name="KeepFailedItemResultsFiles" Value="False"/>
```

The result log files can later be passed to the [DCSExtractResults](#) command line utility to extract information such as the source file used or any errors encountered during conversion. You can search a folder for the results log files using the [DCSCreateFileList](#) utility.

/P - Conversion Profile

This argument is required. The type of file created is controlled by supplying a conversion profile using this switch. The profiles are referenced by passing in the name of the profile XML file, with or without the XML extension. See [Creating and Customizing Profiles](#) for more information about the contents of the profiles, a list of profiles included with Document Conversion Service, and how to create your own.

Examples:

```
/P="PDF 300dpi OptimizedColor"  
/P="TIFF 204x196dpi Monochrome Fax.xml"
```

/D - Define Setting

Individual profile settings can be supplied on the command line using this switch. This switch can be specified multiple times for separate settings. Any settings passed here will override the settings in the profile.

Any name-value pair that can be written in a profile can be passed through this parameter. This includes options to control the conversion settings as well as the behavior of the individual converters as well. See [Creating and Customizing Profiles](#) for more information about the name-value pairs that can be used.

Examples:

These first two are settings that control the converter options, such as what pages to print, and the output that PowerPoint will print.

```
/D="PrintRange:1-5"  
/D="PowerPoint.PrintOptionsOutputType:PrintOutputNotesPages"
```

These two settings control the output file creation options, and would override or add to the settings in the conversion profile passed using the /P switch.

```
/D="Image Options;Fax Resolution:3"  
/D="TIFF File Format;BW compression:Group3-2D"
```

These two settings control where the failed results log files are created and are most often used along with the /FAIL switch to control where the results log files are saved.

```
/D="KeepFailedItemResultsFiles:TRUE"  
/D="UseDateTimeInFailedFolder:FALSE"
```

/E - File Extension Mapping

A file extension mapping profile uses the extension of the source file to determine what converter will be used to convert the file before combining them together. Like the conversion profiles, this file is also an XML file. This switch is optional and an internal default mapping is provided. You would only need to provide this file if you wanted to override the default file extension to converter mappings provided.

Examples:

```
/E="Custom Extension To Converter Map"
```

/W - Wait Time

Use this switch wait to the specified number of seconds for the Document Conversion Service to be running and available to convert and combine documents. If Document Conversion Service is already running the command executes immediately. If the Document Conversion Service is not running in the timeout period specified, the command will return with an error.

If this argument is not specified the command will return immediately with an error if Document Conversion Service is not running.

Example:

```
/W=300
```

/C - Convert on a Remote Computer (DCOM)

If Document Conversion Service is running on a different computer, use this switch to pass the name of the remote computer and the path of a shared location that both computers have access to. Separate the name of the remote computer and the path to the shared folder location with a semi-colon.

When combining remotely, the client redistributable, PNDocConvClientSetup_3.0.exe, must be installed on the computer running this command line utility. The client setup install program is included as part of the Document Conversion Service install and can be found in the **\Samples\Redist** folder in your product installation folder.

Examples:

```
/C="DOC CONV_SERVER;\\DOC CONV_SERVER\DCSREMOTE"
```

/SIL - Smart Inspect Logging File

Smart Inspect Log files are a tracing of the entire conversion process and are not the same as the conversion results log files created when a conversion fails. These logs can be viewed using the SmartInspect Redistributable Console included with Document Conversion Service.

These log files are automatically deleted when conversion succeeds. To keep the log files on success use the custom setting *AlwaysKeepProcessingLoggingFiles* as shown below.

The default location for this file is the TEMP folder. Each logging file is assigned a unique date, time and thread prefix followed by "_PNCombineFolder.sil", such as *2014_09_11_2_38_00_PM_4_PNCombineFolder.sil*.

Use this argument to specify a custom path and optional file name for the SmartInspect logging file (*.sil) created by this utility. The /SIL switch can take a folder, or a path to a filename. If a path without a trailing backslash is provided, the last part of the path is assumed to be a filename.

Note: The double ending backslash used when specifying a folder for the /SIL switch is required for the command line path to be parsed correctly.

/SIL=	Is interpreted as...
"C:\Test\LogFile"	Create the SmartInspect log file as C:\Test\LogFile.sil.
"C:\Test\LogFile\"	Create the SmartInspect log file as C:\Test\LogFile\datetime_PNCombineFolder.sil
"C:\Test\LogFile\ConvertFileCustom.sil"	Create the SmartInspect log file as C:\Test\LogFile\ConvertFileCustom.sil

The following settings can be used to control the creation and naming of the logging file. These settings are all passed using the /D switch.

Custom Setting	Description
RemoveDateTimePrefixOnProcessingLoggingFiles	Pass <i>True</i> to disable the adding of the unique date, time and thread prefix when a custom file name has not been specified in the <i>ConvertFileProcessLoggingPath</i> parameter.
KeepFailedProcessingLoggingFiles	Pass as <i>False</i> to disable the automatic creation of SmartInspect logging files when conversion fails. This setting can be overridden by <i>AlwaysKeepProcessingLoggingFiles</i> .
AlwaysKeepProcessingLoggingFiles	When set to <i>True</i> , the SmartInspect logging files are always created in the %TEMP% or other specified folder for both successful and failed conversions. If set to <i>False</i> , no logging files are created. This setting will override the <i>KeepFailedProcessingLoggingFiles</i> setting.

Examples:

Pass a custom folder and remove the prefix, each run will overwrite the log file C:\PEERNET\Logs\PNCombineFiles.sil.

```
/SIL="C:\PEERNET\Logs\\" /D="RemoveDateTimePrefixOnProcessingLoggingFiles:TRUE"
```

Pass a custom folder and log file name and remove the prefix. Each run will overwrite the logging file C:\PEERNET\Logs\MyLogFile.sil.

```
/SIL="C:\PEERNET\Logs\MyLogFile" /D="RemoveDateTimePrefixOnProcessingLoggingFiles:TRUE"
```

Don't save any SmartInspect log files at all.

```
/D="AlwaysKeepProcessingLoggingFiles:FALSE"
```


/T - Alternate Temp Folder

This is an advanced setting that should not be needed in most cases. When converting files, the conversion tool copies each file and performs the conversion in temporary *staging* and *working* folders created on demand in the default Windows temp folder. When dealing with long path and file names the default folders created can occasionally cause path names that are too long to process. When this happens this switch can be used to set the temporary folder to a shorter path to allow processing.

This setting is overridden if the /C option for remote conversion is being used with its own path to a shared location for conversion.

Examples:

```
/T= "C:\PNTemp\\"
```

/NAME - Sort Files by Name

The file list picked up from the folder is sorted by file name in ascending order (0-9, A-Z).

/CREATED - Sort Files by Creation Date

The file list picked up from the folder is sorted by the files creation date file name in ascending order (0-9, A-Z).

/MODIFIED - Sort Files by Modified Date

The file list picked up from the folder is sorted by the files last modified date file name in ascending order (0-9, A-Z).

/DESC - Sort Files in Descending Order

Use with /NAME, /CREATED or /MODIFIED. Sorts the files in descending order (Z-A, 9-0).

/? - Display Help

When passed as the only argument this switch will display help for this command.

Folder

The folder containing the files to convert.

- If the path to the folder includes spaces it must be enclosed in quotes.
- If the folder doesn't exist, the conversion will fail.

DCSExtractResults

A command line utility to extract information from the results log files. One of the switch arguments must always be specified. If more than one switch is found, the first one is always used. The information extracted is sent to standard out.

```
DCSExtractResults [/s] [/C] [/E] file
```

The results log files are created by the the following command line utilities:

[DCSConvertFile](#)
[DCSConvertFileList](#)
[DCSConvertFolder](#)
[DCSCombineFiles](#)
[DCSCombineFolder](#)

Sample Command Lines

Extract a list of all files created to standard out:

```
DCSExtractResults /C "Document.doc.succeeded.dcsresults"
```

Extract a list of all files created from the *Document.doc.succeeded.dcsresults* log file and sends the information to the console through standard output.

Extract a list of all errors into a text file:

```
DCSExtractResults /E "C:\Test\Output\Document.doc.failed.dcsresults" > "C:\Test\Errors.txt"
```

Extract a list of any errors from the *Document.doc.failed.dcsresults* log file and saves them in the text file C:\Test\Errors.txt.

Extract the source file name of a failed conversion result file:

```
DCSExtractResults /S "C:\Test\Output\Document.doc.failed.dcsresults" >> "C:\Test\Failed.txt"
```

Extracts the source file name from the *Document.doc.failed.dcsresults* file and appends it into the text file C:\Test\CreatedFiles.txt.

Command Line Arguments

Command line switches are not case-sensitive and can be entered in either upper or lower case.

/S - Extract the source file names

Extracts the source file information from the conversion results log file. For [DCSConvertFileList](#) and [DCSCombineFiles](#) this can be more than one file.

/C - Extract the created file names

Extracts the list of created files, if any, from the conversion results log file.

/E - Extract the errors

Extracts the list of errors, if any, from the conversion results log file.

/? - Display Help

When passed as the only argument this switch will display help for this command.

file

The full path to the file to the conversion results log file

DCSCreateFileList

A command line utility to search a folder, and optionally any subfolders and return a list of files matching the search filter specified. The information extracted is sent to standard out.

```
DCSCreateFileList [/R] [F=filter]
                  [/NAME=sort files by name] [/CREATED=sort files by creation date]
                  [/MODIFIED=sort files by name]
                  [/DESC=sort files in descending order (Z-A, 9-0)]
                  searchfolder
```

This utility can be used to search folders for files to send to the command line utilities, or to find the the results log files created by any of command line utilities. The folder search is optimized for speed and efficiency and will return all files that match the filter provided.

Hidden and system files are ignored, and the search pattern filters files based on a regular expression match of the long name of a file. This is different from the Microsoft .Net System.IO.Directory.GetFiles method which returns files based on a check against file names with both the 8.3 file name format and the long file name format, which can cause unexpected file names to be returned.

The file list returned can optionally be sorted by name, date created or date modified. Results are always in ascending order (0-9, A-Z). Pass /DESC to return the results in descending (9-0, Z-A) order.

Sample Command Lines

Searching a folder based on a single file type:

```
DCSCreateFileList /F="*.tif" "C:\Test\Output"
DCSCreateFileList /R /F="*.tif" "C:\Test\Output"
```

Searches the folder C:\Test\Output for all files ending in the pattern *.tif*. Only files with the three letter extension *.tif* will be returned.

The second example will recursively search the folder C:\Test\Output and all subfolders for all files ending on *.tif*.

Search a folder for more than one file type, sort by name:

```
DCSCreateFileList /R /F="*.doc|*.pdf" /NAME "C:\Test\Input"
DCSCreateFileList /R /F="*.doc|*.pdf" /NAME "C:\Test\Input" > C:\Test\InputFileList.txt
```

To search for more than one file type, separate the filter patterns using the pipe (|) character.

This example recursively searches the folder C:\Test\Input for all files ending in the *.doc* or *.pdf* extension. The complete path to all files with only the three letter extension *.doc* and *.pdf* will be returned and sent to the console through standard output.

The second command line shown uses the redirection operator > to redirect the console's standard output into a text file located at C:\Test\InputFileList.txt.

Search the folder C:\Test\Output for all succeeded result log files:

```
DCSCreateFileList /F="*.succeeded.dcsresults" "C:\Test\Output"  
DCSCreateFileList /R /F="*.succeeded.dcsresults" "C:\Test\Output" > "C:\Test\CompletedResults.txt"
```

Searches the folder C:\Test\Output for all results log files that represent completed conversions. The full path to each matching file found is sent to the console through standard output.

The second example does the same as the first except that it recursively searches the folder C:\Test\Output and any subfolders for all results log files that represent completed conversions, not just the root folder. It also uses the redirection operator > to redirect the output into a text file located at C:\Test\CompletedResults.txt.

Search the folder C:\Test\Output for all failed results log files:

```
DCSCreateFileList /F="*.failed.dcsresults" "C:\Test\Output"  
DCSCreateFileList /R /F="*.failed.dcsresults" "C:\Test\Output" > "C:\Test\FailedResults.txt"
```

Searches the folder C:\Test\Output for all results log files that represent failed conversions. The full path to each matching file found is sent to the console through standard output.

The second example does the same as the first except that it recursively searches the folder C:\Test\Output and any subfolders for all results log files that represent failed conversions, not just the root folder. It also uses the redirection operator > to redirect the output into a text file located at C:\Test\FailedResults.txt.

Command Line Arguments

Command line switches are not case-sensitive and can be entered in either upper or lower case.

/R - Include Subfolders (Recurse)

If this switch is used, the subfolders under the folder are included when searching for the list of files that match the filter pattern.

/F - File Filter

Defines the filter that determines what files can be returned, such as using *.pdf to only process PDF files. When this switch is not specified all files (*.*) in the folder are will be returned. Hidden and system files are ignored, and the search pattern filters files based on a regular expression match of the long name of a file.

Multiple filters are combined using the pipe (|) character, such as *.doc|*.pdf to process only Word and PDF files.

/NAME - Sort Files by Name

The file list picked up from the folder is sorted by file name in ascending order (0-9, A-Z).

/CREATED - Sort Files by Creation Date

The file list picked up from the folder is sorted by the files creation date file name in ascending order (0-9, A-Z).

/MODIFIED - Sort Files by Modified Date

The file list picked up from the folder is sorted by the files last modified date file name in ascending order (0-9, A-Z).

/DESC - Sort Files in Descending Order

Use with /NAME, /CREATED or /MODIFIED. Sorts the descending order (Z-A, 9-0).

searchfolder

The full path to the folder in which to start searching.

/? - Display Help

When passed as the only argument this switch will display help for this command.

DCSLicenseDaysLeft

A command line utility to echo to the command line the current license level and the number of days remaining in the subscription period. The information extracted is sent to standard out. This command line tool needs no arguments or parameters.

The command line tool reports as follows:

```
Level Edition; # days
```

A Level I license with 200 days remaining would see the following:

```
Level I Edition; 200 days
```

An expired license reports its expired state:

```
Level I Edition; 0 days [EXPIRED]
```

Document Conversion Service includes several sample profiles for common types of output files for your use. The default set of profiles are installed into the following location:

Default profile location:

C:\ProgramData\PEERNET\Document Conversion Service\Profiles

Custom Profiles

You can use the sample profiles above as a base to edit and create your own custom profiles. Custom profiles can be stored per user in the user's application data folder. Both the local and roaming data folders are searched when looking for user profiles. If a profile is found in a user location, that profile will be used. If no matching profiles are found in the user profile locations, the default profile location is searched.

User profile locations searched in this order:

C:\Users\<user>\AppData\Roaming\Document Conversion Service\Profiles
C:\Users\<user>\AppData\Local\Document Conversion Service\Profiles

When using the PEERNET.ConvertUtility.dll and the command line tools, the full path to a profile stored elsewhere on disk can also be passed instead of the base name of the profile.

See the section [Conversion Settings](#) for information on the contents and structure of the profile files, and the [Name-Value Tables for Conversion Settings](#) for the conversion setting strings to use to get various output formats.

Included Sample Profiles

The profiles included with the Document Conversion Service install are listed below.

See below for [E-discovery specific profiles](#), and [Optical Character Recognition \(OCR\) profiles](#).

Profile Name	Profile Description
Adobe PDF Multipage	Creates Adobe PDF files. The PDF files created using this profile are, where possible, <i>vector</i> PDF files. Vector PDF files are also known as <i>searchable</i> PDF files. The other PDF profiles provided create <i>raster</i> , or non-searchable PDF files. What this profile cannot do is create a vector PDF from an existing raster PDF (scanned PDF) or other image formats such as TIFF or JPEG. A vector PDF is only created if the source document contains text or vector graphics already.
Adobe PDF To RTF	Creates a Rich Text Format document from a PDF file. Does not work with other input file types. only PDF documents.
BMP 100dpi Color	Creates Windows Bitmap images (one image for each page) at 100dpi. Bitmap images are always serialized.
JPEG 60dpi Color JPEG 120dpi Color JPEG 200dpi Color JPEG 300dpi Color JPEG 600dpi Color	Creates color JPEG images (one image for each page) at the dots per inch (dpi) specified. JPEG files are always serialized.
PDF 200dpi OptimizedColor Serialized PDF 300dpi OptimizedColor Serialized	Creates serialized (one file per page) PDF documents at the dots per inch (dpi) specified. Color is optimized per page to reduce file size.
PDF 200dpi OptimizedColor PDF 300dpi OptimizedColor	Creates a multipaged PDF document at the dots per inch (dpi) specified. Color is optimized per page to reduce file size.
PDF A-1b 200dpi OptimizedColor Serialized	Creates serialized (one file per page) PDF/A-1b compliant PDF documents at the dots per inch (dpi) specified. Color is optimized per page to reduce file size.

Profile Name	Profile Description
PDF A-1b 300dpi OptimizedColor Serialized	
PDF A-1b 200dpi OptimizedColor PDF A-1b 300dpi OptimizedColor	Creates a multipaged PDF/A-1b compliant PDF document at the dots per inch (dpi) specified. Color is optimized per page to reduce file size.
Text to Adobe PDF Multipage Text to TIFF 120dpi Monochrome	Profiles for use when converting text files to a searchable PDF or TIFF. Uses the Text - Builtin converter. Page size is auto-detected; see Built-in Text Converter Options for all text settings. Works for all text sizes including wide format (landscape oriented) text files such as those generated on mainframe systems or other reporting systems.
Text to A3 sized TIFF 120dpi Monochrome Text to A3 sized PDF 120dpi Monochrome	<i>Deprecated; use Text to Adobe PDF Multipage or Text to TIFF 120dpi Monochrome with the Text - Builtin converter instead.</i> Profiles for use when converting text files using Word to a specific size of paper. These profiles target wide format (landscape oriented) text files such as those generated on mainframe systems or other reporting systems.
TIFF 120dpi Color LowJPEG TIFF 150dpi Color LowJPEG TIFF 200dpi Color LowJPEG TIFF 300dpi Color LowJPEG TIFF 600dpi Color LowJPEG	Creates multipaged color TIFF images at the dots per inch (dpi) specified. Images are compressed using low quality JPEG compression. This can give a smaller file size but a lower quality image.
TIFF 120dpi Color HighPEG TIFF 150dpi Color HighPEG TIFF 200dpi Color HighPEG TIFF 300dpi Color HighPEG TIFF 600dpi Color HighPEG	Creates multipaged color TIFF images at the dots per inch (dpi) specified. Images are compressed using high quality JPEG compression. This can give a higher quality image but also a larger size file.
TIFF 120dpi Grayscale TIFF 150dpi Grayscale TIFF 200dpi Grayscale TIFF 300dpi Grayscale TIFF 600dpi Grayscale	Creates multipaged grayscale TIFF images at the dots per inch (dpi) specified.
TIFF 120dpi OptimizedColor TIFF 150dpi OptimizedColor TIFF 200dpi OptimizedColor TIFF 300dpi OptimizedColor TIFF 600dpi OptimizedColor	Creates a single multipage TIFF image at the dots per inch (dpi) specified. Color is optimized per page to reduce file size. File is compressed using Group 4 compression for monochrome and LZW for all other color types.
TIFF 200dpi OptimizedColor HighJPEG	Creates a single multipage TIFF image at the dots per inch (dpi) specified. Color is optimized per page to reduce file size. File is compressed using Group 4 compression for monochrome and high quality JPEG compression for all other color types.
TIFF 200dpi Monochrome Serialized	Creates serialized (one file per page) black and white TIFF images at 200dpi.
TIFF 200dpi Monochrome	Creates a single multipage black and white TIFF image at 200dpi.
TIFF 204x196dpi Monochrome Fax	Creates a single multipage black and white fax format TIFF image at 204 x 196dpi.
TIFF 204x196dpi Monochrome Fax ReverseBitOrder	Creates a single multipage black and white Group 4 fax format TIFF image at 204 x 196dpi with a reverse bit order of least significant bit to most significant bit (LSB2MSB). Often needed for fax boards.

Profile Name	Profile Description
TIFF 204x196dpi Monochrome Fax Group3 256GreyPalette	Creates a single multipage Group 3 fax format TIFF image at 204 x 196dpi using a grayscale palette.
TIFF 204x196dpi Monochrome Fax Group3 256GreyPalette ReverseBitOrder	Creates a single multipage Group 3 fax format TIFF image at 204 x 196dpi using a grayscale palette with a reverse bit order of least significant bit to most significant bit (LSB2MSB).
TIFF 204x196dpi Monochrome Fax Compatible with FCC	Created fax TIFF images matching the format created by the Fax(TIFF) profile used in PEERNET File Conversion Center. Provided for use by clients migrating from File Conversion Center to Document Conversion Service.
TIFF 300dpi Allow Javascript PDF	This profile is the same as the <i>TIFF 300dpi Optimized Color</i> above but also enables the processing of Javascript, if present, in PDF files when they are converted using this profile.
TIFF 300dpi Color Fax	Creates a single multipage color fax format TIFF image at 300dpi.
TIFF 300dpi OptimizedColor ExtractText Serialized	Creates serialized (one file per page) TIFF images at 300dpi. Color is optimized per page to reduce file size. Text content, if available, is extracted and saved as separate files with the same base name as the output images.
TIFF 300dpi OptimizedColor ExtractText	Creates a single multipage TIFF image at 300dpi. Color is optimized per page to reduce file size. Text content, if available, is extracted and saved as a separate file with the same base name as the output image.
TIFF 300dpi OptimizedColor Serialized	Creates serialized (one file per page) TIFF images at 300dpi. Color is optimized per page to reduce file size.
TIFF 300dpi OptimizedColor SplitByPageCount	Creates a sequence of multipaged 300 dots per inch TIFF images. A new file in the sequence is started based on the page count set by the <i>SplitFileEveryNPages</i> setting. When auto-splitting files, serialized naming profile is always used to name each file in the sequence.
TIFF 300dpi OptimizedColor SplitByFileSize	Creates a sequence of multipaged 300 dots per inch TIFF images. A new file in the sequence is started when the current file exceeds the file size set by the <i>SplitFileSizeThresholdInBytes</i> setting. When auto-splitting files, serialized naming profile is always used to name each file in the sequence.
Text to A3 sized TIFF 120dpi Monochrome Text to A3 sized PDF 120dpi Monochrome	Profiles for use when converting text files in Word to a specific size of paper. These profiles target wide format (landscape oriented) text files such as those generated on mainframe systems or other reporting systems.

Optical Character Recognition (OCR) Profiles	Profile Description
Adobe PDF OCR to Searchable Adobe PDF OCR to Searchable Serialized	These profiles can be used to perform OCR when converting to PDF with any of the PEERNET built-in converters installed with Document Conversion Service. Optical Character Recognition (OCR for short), searches for and recognizes text (characters) on scanned pages or images and extracts it as digital text included in the PDF file.
E-Discovery Profiles	Profile Description
eDiscovery - Excel - PDF 300dpi Convert Charts Only eDiscovery - Excel - TIFF 300dpi Convert Charts Only	For use with Excel documents, these profiles will print only the embedded charts and any chart tabs in the document.
eDiscovery - Excel - PDF 300dpi Show Formulas eDiscovery - Excel - TIFF 300dpi Show Formulas	For use with Excel documents, these profiles will print any formulas from any cells as a comment at the end of each sheet. If a comment already exists, the formula is inserted before the existing text. For Excel documents, a tracked changes history sheet is created if tracking is enabled, background colors are removed, text is changed to black and conditional formatting is removed.
eDiscovery PDF 300dpi AutoField Replace eDiscovery TIFF 300dpi AutoField Replace	For use with Word, Excel and PowerPoint e-discovery, these profiles will show all data in the documents and where possible, replace any auto data, time and file fields in headers, footers, and in the case of Excel, in cells too. For Excel documents, a tracked changes history sheet is created if tracking is enabled, background colors are removed, text is changed to black and conditional formatting is removed.
eDiscovery PDF 300dpi Monochrome Fit On Page eDiscovery TIFF 300dpi Monochrome Fit On Page	For use with Word, Excel and PowerPoint e-discovery, these profiles will show all data in the documents. The output created is black and white. For Excel documents, each sheet is fit to a single output page, a tracked changes history sheet is created if tracking is enabled, background colors are removed, text is changed to black and conditional formatting is removed.
eDiscovery PDF 300dpi Span Pages eDiscovery TIFF 300dpi Span Pages	For use with Word, Excel and PowerPoint e-discovery, these profiles will show all data in the documents. For Excel documents, tracked changes history sheet is created if tracking is enabled, background colors are removed, text is changed to black and conditional formatting is removed.

Conversion Settings

Conversion settings are used to describe the output created by Document Conversion Service and consist of a collection of name-value pairs. These settings can also be used to control the behavior of the individual converters, such as configuring Word to pass a password or telling Excel to ignore the print areas when printing worksheets.

Conversion Profiles

When using the command line utilities the settings are passed by supplying the name of a *profile file*, a structured XML file that contains the list of settings. Below is a sample command line using a profile file named *TIFF 300dpi Optimized Color.xml*, followed by a listing of the XML file itself. Note that the *.xml* extension is not needed when using the command line utilities. Several sample profiles are included for your use, or to use as a base to [customize](#) to your needs.



Passing setting using a profile

```
DCSConvertFile /P="TIFF 300dpi OptimizedColor" /NE "C:\Test\File.pdf"
```



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="TIFF 300dpi OptimizedColor"
  Description="Creates a single TIFF image at 300dpi.">
  <Settings>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name="Save;Output File Format" Value="TIFF Multipaged"/>
    <add Name="Save;Color reduction" Value="Optimal"/>
    <add Name="Save;Dithering method" Value="Halftone"/>

    <!-- TIFF Compression Options -->
    <add Name="TIFF File Format;BW compression" Value="Group4"/>
    <add Name="TIFF File Format;Color compression" Value="LZW RGB"/>
    <add Name="TIFF File Format;Indexed compression" Value="LZW"/>
    <add Name="TIFF File Format;Greyscale compression" Value="LZW"/>
    <add Name="JPEG File Format;Color compression" Value="Medium Quality"/>
    <add Name="JPEG File Format;Greyscale compression" Value="High Quality"/>
    <add Name="Image Options;Fax" Value="0"/>

  </Settings>
</Profile>
```

Name-Value Tables for Conversion Settings

The table below lists the different conversion settings separated out into categories with a description of the settings available in each. Click the link for that category to view all available settings for that option.

Options	Description of Settings
General Converter Options	These are general options that can be applied to the conversion process itself or to all converters.
Built-in Converter OCR Options	These options can be used to perform OCR when converting to PDF with any of the PEERNET built-in converters installed with Document Conversion Service. Optical Character Recognition (OCR for short), searches for and recognizes text (characters) on scanned pages or images and extracts it as digital text.
Built-in Cadd Converter Options	Convert DWF, DWFX, PLT and GBX files using the builtin Cadd conversion utility. This converter does not require any third-party applications.
Built-in Image Converter Options	The built-in image converter is a faster, more efficient converter designed to replace our original Image Converter. This converter has no options.
Built-in PDF Converter Options	Convert PDF files using the built-in PDF conversion utility instead of Adobe Reader. This converter has no options.
Built-in Text Converter Options	Convert text files using this built-in conversion utility instead of Microsoft Word. This converter has options for font name and size. It auto-detects page size but also offers font and page size settings.
Word Converter Options	These options are specific to the behavior of the Word converter.
Excel Converter Options	These options are specific to the behavior of the Excel converter.
PowerPoint Converter Options	These options are specific to the behavior of the PowerPoint converter.
Ghostscript Converter Options	These options are specific to the behavior of the Ghostscript converter.
Image Converter Options	These options are specific to the behavior of the Image converter.
OutsideIn AX Options	These options are specific to the behavior of the OutsideIn converter.
Advanced Features	Advanced settings such as custom paper size and text extraction.
Advanced File Naming	Settings to configure the file naming profiles (preset file naming schemes) for multipaged, multipaged with JobID, serialized and serialized with JobID.
Devmode settings	Resolution (DPI), page size and color mode settings.
Image Options	Image output options such as creating fax mode images and page rotation settings.
JPEG File Format	Compression settings for color and greyscale JPEG images.
PDF File Format	PDF file format settings for compression, content encoding and PDF/A-1b compliant PDF files.

Options	Description of Settings
PDF Security	PDF encryption and file permissions.
Processing	Settings to adjust the image during conversion such as trimming, cropping, copying to a new page size, resampling and brightness adjustment.
Save	Settings for output file format, color reduction, dithering and file name prompting.
TIFF File Format	Compression settings for black and white, color, indexed and greyscale TIFF images.
Endorsement Options	Endorsements are header and footer information that can be stamped onto each page of the output created by Document Conversion Service.
Watermark Stamping	Settings to create a text watermark diagonally across the page.

File Extension to Converter Mapping

The file extension of each file is used to determine what converter is used when Document Conversion Service converts that file.

When using the command line tools to convert files, a default file extension mapping profile, *File Extension To Converter Map.xml*, is used to determine this mapping. This file can be edited and file extensions can be added, removed and changed as needed.

If desired, the file itself can be copied and renamed and the new mapping file passed to the PEERNET.ConvertUtility methods or the command line tools as needed.

A simpler approach is to customize the file extension mapping by adding the setting into a profile file. This allows you to set the file extension mapping at a file level instead of at the application level. Any file extension mappings found in a profile will override the settings in the base *File Extension To Converter Map.xml* file.

A common use of this would be to have a profile that uses the PEERNET Passthrough Converter to skip processing TIFF files, or one that uses Ghostscript to process PDF files instead of Adobe Reader.

Customizing the File Extension Mapping Profile

File mapping profiles are stored in the same location as the conversion profiles. The default file extension mapping profile, *File Extension To Converter Map.xml*, is installed as part of Document Conversion Service. The difference between a conversion profile and a mapping profile is detected using the Type attribute on the Profile element. It is 0 for a conversion profile and 1 for a file extension mapping profile.

The mapping consists of the extension (the suffix of the file name past the last dot or period in file's name) and a semi-colon separated list of converter names. There are two things to remember when modifying this file:

1. Each file extension can only be listed once.
2. The file extensions must be added in lower case and must include both the dot (.) and the extension.

In some cases the file extension may only have one converter associated with it. Others, such as PDF which can be converted using either our Adobe PDF - Builtin converter, Adobe Reader, Ghostscript or Outside-In AX, can potentially have more than one converter, in order of preference, associated with it. The code sample below shows a small snippet of the file mapping in the provided file mapping profile.

You can edit the File Extension Mapping profile in the DCS Editor by going to **DCS Dashboard - DCS Settings - Edit File Extension Map**. You can also open the XML file using your favorite text or code editor.



Code Sample - File Extension to Converter Mapping

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="1"
  DisplayName="File Extension To Converter Map"
  Description="Maps file extensions to the converter to use for that document.">
  <Settings>
    <add Name=".doc" Value="Microsoft Word;Outside-In AX"/>
    <add Name=".docx" Value="Microsoft Word;Outside-In AX"/>
    ...
    <add Name=".xlsx" Value="Microsoft Excel;Outside-In AX"/>
    <add Name=".xlsm" Value="Microsoft Excel;Outside-In AX"/>
    ...
    <add Name=".pdf" Value="Adobe PDF - Builtin;Adobe Acrobat Reader;Ghostscript;Outside-In AX"/>
    ...
  </Settings>
</Profile>
```

The table below lists the available converters and their default file extensions.

Converter Name	Supported Document Types
Adobe Acrobat Reader	Adobe PDF Documents (*.pdf)
Adobe PDF - Builtin	Adobe PDF Documents (*.pdf)
Autodesk Design Review	Design Review Drawings (*.dwf) AutoCAD Drawings (*.dwg)
Microsoft Excel	Excel Workbooks (*.xlsx, *.xlsm, *.xls) Excel Templates (*.xltx, *.xltm, *.xlt) Excel Binary Workbook (*.xlsb)
Cadd - Builtin	Design Review Drawings (*.dwf, *.dwfx) HPGL Plot files (*.plt) Gerber RS-274D, Gerber RS-274X, Gerber X2 (*.gbx)
Ghostscript	Postscript Files (*.ps) Encapsulated Postscript Files (.eps) Adobe PDF Documents (*.pdf)
Image - Builtin (replaces PEERNET Image Converter and PEERNET WIC Image Converter)	JPEG images (*.jpg) TIFF images (*.tif) High Efficiency Image Files (*.heif, *.heic) Google WebP Images (*.webp) AVIF Images (*.avif) Windows Bitmap images (*.bmp) ZSoft PCX images (*.pcx) ZSoft DCX images (*.dcx) CServe Portable Network Graphics images (*.png) Graphics Interchange Format image files (*.gif) Icon Format (*.ico) Windows Media Photo images (*.wdp, *.hdp, *.jxr) ImageMagick images (100+ image formats)
PEERNET Image Converter	JPEG images (*.jpg) TIFF images (*.tif) Windows Bitmap images (*.bmp) ZSoft PCX images (*.pcx) ZSoft DCX images (*.dcx) CServe Portable Network Graphics images (*.png) Graphics Interchange Format image files (*.gif) Icon Format (*.ico) Windows Media Photo images (*.wdp, *.hdp, *.jxr)
PEERNET WIC Image Converter	Icon Format (*.ico) Windows Media Photo images (*.wdp, *.hdp, *.jxr) Works with other Windows Imaging Component (WIC) third-party add-ons such as: DjVu Shell Extension Pack (*.djvu) FastPicture Viwer Codec Pack adds support for over 45+ image formats and over 500 raw digital camera formats
Internet Explorer	HTML Files (*.htm, *.html) Secure HTML (*.shtm, *.shtml) Web Archive (*.mht)
Microsoft Outlook	Outlook Message Files (*.msg) Outlook Templates (*.oft)
Outside-In AX	Oracle Outside In Viewer Technology (ActiveX) supports over 500 common file formats; see the documentation that came with your Outside In Technology product.

Converter Name	Supported Document Types
Microsoft PowerPoint	PowerPoint Presentations (*.pptx, *.pptm, *.ppt) PowerPoint Shows (*.ppsx, *.ppsm, *.pps) PowerPoint Templates (*.potx, *.potm, *.pot)
Microsoft Publisher	Publisher Files (*.pub)
Text - Built-in	Text Files (*.txt) Logging Files (*.log) Source Files(*.c, *.cpp, *.cs, etc.) Config and INI Files (.config, *.ini) PowerShell (*.ps1) Batch Files (*.cmd, *.bat) Any other text-based file
Microsoft Visio	Visio Drawings (*.vsd)
Microsoft Word	Word Documents (*.docx, *.docm, *.doc) Word Templates (*.dotx, *.dotm, *.dot) Rich Text Documents (*.rtf) Plain Text Files (*.txt) Plain Text Log Files (*.log)
Microsoft XPS	XPS Documents (*.xps) Open XPS Documents (*.oxps)
PEERNET Passthrough	Any file type. Passes any file matching the extension through the system without converting.

General Converter Options

These options can be used with any of the converters installed with Document Conversion Service. Table values in **bold text** are the default value for that setting.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="TIFF 300dpi First3PagesOnly "
  Description ="Prints only the first three pages.">
  <Settings>

    <!-- Print first three pages only -->
    <add Name="PageRange" Value="1-3"/>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name="Save;Output File Format" Value="TIFF Multipaged"/>
    ...

  </Settings>
</Profile>
```

Conversion Settings

Name: PageRange

The page numbers and page ranges to include in the output file. Separate each number and range with a comma. For example, "1, 3, 5-7" prints page 1 and 3 and pages 5 through 7. Numbers in the page range exceeding the page count of the source document are ignored.

Values: The string representing the page range.

Name: MaxSpooledPagesAllowed

Sets the maximum number of pages that are allowed to be printed/spooled. Documents larger than this set page limit will not convert.

Values: The string representing the maximum number of pages allowed.

Name: MaxSpooledPagesGreaterThanPageCount

Sets the maximum number of spooled pages greater than the document page count that is allowed to be printed. Documents larger than this set page limit will not convert. This is often used to manage a single extra page created by duplex printing forced by the document. It can also occur with mail merge documents and PDF files that use Javascript. For PDF files, use the **Adobe.PDF.Javascript.Enable** setting instead.

When not set (default), or set to 0, the conversion will fail if the number of spooled pages is greater than the document page count.

Values: The string representing the maximum number of pages allowed.

Conversion Settings

Name: **NormalizeFileNames**

When *true*, file names passed in will be checked for normalization and normalized when necessary. This means that the new output file name, if not specified, will be the normalized filename. The default is to not normalize the filename.

This is needed for foreign file name where some international characters are represented using diacritics. A diacritic is a *glyph* added to a letter; they are used to change the sound of the letter to which they are added. Some examples of a diacritic are the accent grave (') and acute (') in the French language.

Values: Pass *true* to normalize file names if necessary.

Name: **SecondsToWaitForRunningConversionService**

Applies only when using the command line tools (/D switch) and the PEERNET.ConvertUtility methods.

The Document Conversion Service must be running, either locally or on a remote computer for files or folders of files to be converted. If it is not running the PEERNET.ConvertUtility methods or command line tools it will all return immediately with an error. To wait for Document Conversion Service to be running instead of failing to convert the files, use this setting to pass the desired wait timeout value down. If Document Conversion Service hasn't started after waiting the supplied amount of time, an error is returned.

Values: The number of seconds to wait for Document Conversion Service to be running and ready to convert files.

Name: **KeepFailedItemResultsFiles**

Applies only when using the command line tools (/D switch) and when passing custom settings to the PEERNET.ConvertUtility methods.

By default when a conversion fails, a results file ending with .failed.dcsresults for the file that failed will be created in a .failed folder. To suppress the automatic creation of these files pass this setting as *true*. When using the PEERNET.ConvertUtility methods, the resultant items that are returned will contain the path to the results file.

Values: Pass *true* to suppress the creation of these files.

Name: **FailedFolder**

Applies only when passing custom settings to the PEERNET.ConvertUtility methods.

By default when a conversion fails, a results file ending with .failed.dcsresults for the file that failed will be created in a .failed folder. Specifying a folder for this custom setting will override the default use of the .failed folder and store the failed results log files if the specified folder.

Values: Pass the path to the folder in which to store the failed conversion results files.

Conversion Settings

Name: **AlwaysKeepProcessingLoggingFiles**

Applies only when using the command line tools (/D switch) and the PEERNET.ConvertUtility methods.

By default a Smart Inspect console logging file (*.sil) is always created when a conversion runs. If the conversion is successful, the log file is normally deleted. If it fails, it is kept and copied to the Windows temp folder. To always keep this file, pass this setting as true. Overrides the variable **KeepFailedProcessingLoggingFiles**. When using the PEERNET.ConvertUtility methods, the results items that are returned will contain the path to the results file.

Values: Pass *true* to always keep the logging file.

Name: **KeepFailedProcessingLoggingFiles**

Applies only when using the command line tools (/D switch) and the PEERNET.ConvertUtility methods.

By default when a conversion fails, the Smart Inspect console logging file (*.sil) created as part of the conversion process is kept and copied to the Windows temp folder. To have these files deleted even when the conversion fails, pass this setting as *true*. When using the PEERNET.ConvertUtility methods, the results items that are returned will contain the path to the results file.

Values: Pass *true* to delete these files when the conversion is finished even if the conversion has failed.

Name: **UseCompressedDateTimeFormat**

Applies only when using the command line tools (/D switch) and the PEERNET.ConvertUtility methods.

Controls the formatting of the name of the date and time subfolder used internally by the conversion utility in the staging and working folders for file conversion, as well as in naming the internal logging files (*.sil). This setting would only need to be altered if you are dealing with very long folder and file path names that exceed the 255 character path limit, as a way of reducing the internally created paths so that they do not exceed the maximum path length.

When set to *FALSE*, or not provided, the folder name follows the pattern '2016_03_31_2_38_46_PM'. The compressed format is shorter, and uses a 24-hour time format, giving a folder following the pattern '20160331143846'.

Values: Pass *true* to use the shorter, numerical format.

Built-in Converter OCR Options

These options are used when converting to PDF with any of the PEERNET built-in converters installed with Document Conversion Service. These settings do not apply to any other output format.



Caution

This feature is not supported on Microsoft® Windows Server 2008 R2 and Microsoft® Windows 7.

Optical Character Recognition (OCR for short), searches for and recognizes text (characters) on scanned pages or images and extracts it as digital text. When recognizing text, the OCR engine has to know which languages to look for on the page. OCR works by analyzing the patterns, shapes, and curves of the text characters on the page and matching them to predefined information for different characters in each language.

OCR will increase the processing time for file conversion. Outside factors such as image quality, the font used, and any image background on the pages will all affect the validity of the OCR results.

They are used by the following converters:

- Built-in PDF Converter
- Built-in Image Converter
- Built-in Cadd Converter

Table values in **bold** text are the default value for that setting.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="Adobe PDF OCR Searchable"
  Description ="Converts to OCR (searchable) PDF.">
  <Settings>

    <add Name ="ConverterPlugIn.PNBuiltinsOCRPDF.Enabled" Value="1"/>
    <add Name ="ConverterPlugIn.PNBuiltinsOCRPDF.Languages" Value="eng+fra"/>
    <add Name ="ConverterPlugIn.PNBuiltinsOCRPDF.FirstPageOnly" Value="0"/>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name="Save;Output File Format" Value="Adobe PDF Multipaged"/>
    ...

  </Settings>
</Profile>
```

Conversion Settings -PNBuiltinCaddConverter	
Name:	ConverterPlugIn.PNBuiltinsOCRPDF.Enabled
	Enable this setting to run OCR on your pages and create a searchable PDF file. Running OCR does slow down the conversion process. Does not apply to any other output format.
Values:	String value; 1 to enable OCR, 0 to disable.

Conversion Settings -PNBuiltinCaddConverter**Name:** **ConverterPlugIn.PNBuiltinsOCRPDF.Languages**

A string representing the languages to recognize on the page. English is preselected by default. You must have at least one language selected for OCR.

Values: Select multiple languages by separating each language code with a plus (+) sign. For example, "eng+fra+deu" will recognize English, French and German text on your pages.

- Arabic(ara)
- **English(eng)**
- French(fra)
- German(deu)
- Hebrew(heb)
- Hindi(hin)
- Italian(ita)
- Spanish(spa)

Additional languages can be downloaded and added to Document Conversion Service if necessary.

Name: **ConverterPlugIn.PNBuiltinsOCRPDF.FirstPageOnly**

Allows you to only run the OCR process on the first page of every file.

Values: String value; 1 to process only the first page, 0 to process all the pages in the document.

Adding Languages

Document Conversion Service comes with files to support recognizing Arabic, English, French, German, Hebrew, Hindi, Italian, and Spanish. You can download additional language files or complete sets of language files from [Traineddata Files for Tesseract](#).

To add them to Document Conversion Service, copy the desired *.traineddata files into the following folder:

%PROGRAMDATA%\PEERNET\Document Conversion Service\tessdata

Built-in PDF Converter Options

The Built-in PDF Converter replaces the [Adobe Reader converter](#). It eliminates the need to install Adobe Reader to convert PDF files. Unlike when converting using the Adobe Reader converter, files are opened and converted without printing, resulting in faster conversion.

**Caution**

This feature is not supported on Microsoft® Windows Server 2008 R2 and Microsoft® Windows 7.

This converter has no individual options.

Built-in Text Converter Options

This converter handles processing text-based files, including wide-format text files. It handles text-based log files, batch files, source code files, and many others.



Caution

This feature is not supported on Microsoft® Windows Server 2008 R2 and Microsoft® Windows 7.

Text files may or may not contain form feeds. A form feed is a special character that causes any text after it to start on a new page. When your text document has form feeds, the text encoding, font size, and page size used to position the page contents are auto-detected.

A document with no form feed characters is considered a single page. The number of characters and lines, along with the font and font size, is used to calculate a page size. The text is split into pages using the calculated page size. Text files that commonly have no form feeds are source code files, XML files, log files, and similar.

There are two sets of options - one for [text files with form feeds](#) and one for [text documents without form feeds](#).

Table values in bold text are the default value for that setting.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="TIFF 300dpi Custom Text"
  Description="Converts Text files using a custom font.">
  <Settings>
    <add Name
      ="ConverterPlugIn.PNBuiltinTextConverterSettingsContentWithFormFeeds.FontName"
      Value="Consolas"/>
    <add Name
      ="ConverterPlugIn.PNBuiltinTextConverterSettingsContentWithFormFeeds.FontSizeInPoints"
      Value="11"/>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name="Save;Output File Format" Value="TIFF Multipaged"/>
    ...
  </Settings>
</Profile>
```

This table lists the options for handling text files **with form feeds** (page eject). A form feed automatically causes any text after it to start on a new page. For options for text files **without form feeds**, see [this table](#).

Conversion Settings -PNBuiltinTextConverter - Text With Form Feeds Options

Name: **ConverterPlugIn.PNBuiltinTextConverterSettingsContentWithFormFeeds.FontName**

Choose which font to use for the text in your document. Choose a fixed-width (fixed pitch) font for best results when your text files contain tables and column of data when alignment is important. If the font specified does not exist, or the name is spelled incorrectly, the default of **Courier New** is used.

Values: String value of the font name. The default is **Courier New**.

Conversion Settings -PNBuiltinTextConverter - Text With Form Feeds Options

Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentWithFormFeeds.FontSizeInPoints
	<p>The size of the font is in points. Each point is 1/72 of an inch. A font size of 10 points means each character is approximately 1/14 of an inch tall.</p> <p>If you change the font and/or the font size, and you are word-wrapping the text, you may need to adjust the MaxCharactersPerLine to account for your document content and chosen font.</p>
Values:	String value of the size of the font in points. The default is 10 . The smallest size allowed is 6 , and largest, 72 .
Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentWithFormFeeds.LineWrapMode
	<p>Line wrapping is the automatic shifting of text to a new line when we have reached the maximum number of characters per line. The maximum number of characters per line depends on any font, font size, paper size, and maximum character settings you have set.</p>
Values:	<p>None - Do not line wrap. When not line wrapping, the paper sizany text that occurs past the maximum number of characters per line is truncated.</p> <p>Plain - wraps the text at the calculated number of characters per line, with no attempt to backtrack to find the last word break.</p> <p>Word - wraps the text at the calculated number of characters per line, backtracking to find the last word break when necessary.</p>
Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentWithFormFeeds.MaxCharactersPerLine
	<p>Specifies the maximum number of characters per line when <i>LineWrapMode</i> is set to <i>Plain</i> or <i>Word</i>. The desired maximum number of characters per line depends on any font, font size, and paper size you have set. You may need to adjust this if you change the font and/or the font size.</p>
Values:	String value of the maximum number of characters per line. The default is 83 .
Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentWithFormFeeds.PageSizeDetectionMode
	<p>Choose to auto-detect the page size needed for your text document, size the content to fit a specified page size, or always use a specified page size.</p>
Values:	<p>AutoDetect - Scans the text document to determine the number of characters per line and lines per page. The number of characters and lines, and form feed characters, along with the <i>FontName</i> and <i>FontSizeInPoints</i>, is used to calculate the page size needed. Using this page size, we search the Document Conversion Service printer's forms list for a paper size that is the closest match without being smaller. If we cannot find a matching paper size, we use the settings for <i>PageSize_DefaultPageWidthInPoints</i> and <i>PageSize_DefaultPageHeightInPoints</i>.</p> <p>SizeToFit - Scans the text document to determine the number of characters per line and lines per page. The number of characters and lines, along with the font and font size, is used to calculate the page size. Unlike AutoDetect, we use this calculated page size as the paper size. There is no attempt to match to a standard paper size.</p> <p>UseSpecified - Always uses the page size set using <i>PageSize_DefaultPageWidthInPoints</i> and <i>PageSize_DefaultPageHeightInPoints</i>. Text that exceeds the width or height of the page can get cut off.</p>

Conversion Settings -PNBuiltinTextConverter - Text With Form Feeds Options

Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentWithFormFeeds.PageSize_AutoDetect_MinPageWidthInPoints
	The minimum width of an auto-detected page size. The page will not be smaller than this width, but it can be wider. This is entered in points. There are 72 points per inch, so a default width of 576 is 8 inches.
Values:	String value of the minimum page width in points. The default is 576 , or 8 inches.
Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentWithFormFeeds.PageSize_AutoDetect_MinPageHeightInPoints
	The minimum height of an auto-detected page size. The page will not be shorter than this width, but it can be taller. There are 72 points per inch, so a height of 576 is 8 inches.
Values:	String value of the minimum page height in points. The default is 576 , or 8 inches.
Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentWithFormFeeds.PageSize_DefaultPageWidthInPoints
	Sets the desired page width in points. This setting is used if <i>PageSizeDetectionMode</i> is set to <i>UseSpecified</i> , or if it is set to <i>AutoDetect</i> and it does not match an appropriate page size. There are 72 points per inch, so the default width of 612 is 8.5 inches.
Values:	String value of the desired page width in points. The default is 612 , or 8.5 inches.
Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentWithFormFeeds.PageSize_DefaultPageHeightInPoints
	Sets the desired page height in points. This setting is used if <i>PageSizeDetectionMode</i> is set to <i>UseSpecified</i> , or if it is set to <i>AutoDetect</i> and it does not match an appropriate page size. There are 72 points per inch, so the default height of 792 is 11 inches.
Values:	String value of the desired page height in points. The default is 792 , or 11 inches.
Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentWithFormFeeds.PageSize_LeftMarginInPoints
	Sets the width of the left margin. A margin is the space between the edge of the page and the main body of text. There are 72 points per inch, so the default width of 18 is 0.25 inches.
Values:	String value of the desired left margin width in points. The default is 54 , or 0.75 inches.
Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentWithFormFeeds.PageSize_RightMarginInPoints
	Sets the width of the right margin. A margin is the space between the edge of the page and the main body of text. There are 72 points per inch, so the default width of 18 is 0.25 inches.
Values:	String value of the desired right margin width in points. The default is 54 , or 0.75 inches.

Conversion Settings -PNBuiltinTextConverter - Text With Form Feeds Options

Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentWithFormFeeds.PageSize_TopMarginInPoints
	Sets the height of the top margin. A margin is the space between the edge of the page and the main body of text. There are 72 points per inch, so the default height of 18 is 0.25 inches.
Values:	String value of the desired top margin height in points. The default is 72 , or 1 inch.
Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentWithFormFeeds.PageSize_BottomMarginInPoints
	Sets the height of the bottom margin. A margin is the space between the edge of the page and the main body of text. There are 72 points per inch, so the default height of 18 is 0.25 inches.
Values:	String value of the desired bottom margin height in points. The default is 72 , or 1 inch.
Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentWithFormFeeds.TabStopEveryNSpaces
	Replaces any tab stops in the text document with the given number of spaces (columns).
Values:	String value of the number of spaces to replace the tab stop with. The default is 8 spaces.

This table lists the options for handling text files **without form feeds** (page ejects). When a text document does not have form feeds, the settings **PageSize_DefaultPageWidthInPoints** and **PageSize_DefaultPageHeightInPoints** are used to determine the paper size and when a new page begins. For options for text files **with form feeds**, see [this table](#).

Conversion Settings -PNBuiltinTextConverter - Text With No Form Feeds Options

Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentNoFormFeeds.FontName
	Choose which font to use for the text in your document. Choose a fixed-width (fixed pitch) font for best results when your text files contain tables and column of data when alignment is important. If the font specified does not exist, or the name is spelled incorrectly, the default of Courier New is used.
Values:	String value of the font name. The default is Courier New .
Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentNoFormFeeds.FontSizeInPoints
	The size of the font is in points. Each point is 1/72 of an inch. A font size of 10 points means each character is approximately 1/14 of an inch tall.
	If you change the font and/or the font size, and you are word-wrapping the text, you may need to adjust the MaxCharactersPerLine to account for your document content and chosen font.
Values:	String value of the size of the font in points. The default is 10 . The smallest size allowed is 6 , and largest, 72 .

Conversion Settings -PNBuiltinTextConverter - Text With No Form Feeds Options

Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentNoFormFeeds.LineWrapMode
	Line wrapping is the automatic shifting of text to a new line when we have reached the maximum number of characters per line. The maximum number of characters per line depends on any font, font size, paper size, and maximum character settings you have set.
Values:	<p>None - Do not line wrap. When not line wrapping, any text that occurs past the maximum number of characters per line is truncated.</p> <p>Plain - wraps the text at the calculated number of characters per line, with no attempt to backtrack to find the last word break.</p> <p>Word - wraps the text at the calculated number of characters per line, backtracking to find the last word break when necessary.</p>
Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentNoFormFeeds.MaxCharactersPerLine
	Specifies the maximum number of characters per line when <i>LineWrapMode</i> is set to <i>Plain</i> or <i>Word</i> . The desired maximum number of characters per line depends on any font, font size, and paper size you have set. You may need to adjust this if you change the font and/or the font size.
Values:	String value of the maximum number of characters per line. The default is 83 .
Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentNoFormFeeds.PageSizeDetectionMode
	Choose to auto-detect the page size needed for your text document, size the content to fit a specified page size, or always use a specified page size.
Values:	<p>AutoDetect - Scans the text document to determine the number of characters per line and lines per page. The number of characters and lines, and form feed characters, along with the <i>FontName</i> and <i>FontSizeInPoints</i>, is used to calculate the page size needed. Using this page size, we search the Document Conversion Service printer's forms list for a paper size that is the closest match without being smaller. If we cannot find a matching paper size, we use the settings for <i>PageSize_DefaultPageWidthInPoints</i> and <i>PageSize_DefaultPageHeightInPoints</i>.</p> <p>SizeToFit - Scans the text document to determine the number of characters per line and lines per page. The number of characters and lines, along with the font and font size, is used to calculate the page size. Unlike AutoDetect, we use this calculated page size as the paper size. There is no attempt to match to a standard paper size.</p> <p>UseSpecified - Always uses the page size set using <i>PageSize_DefaultPageWidthInPoints</i> and <i>PageSize_DefaultPageHeightInPoints</i>. Text that exceeds the width or height of the page can get cut off.</p>
Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentNoFormFeeds.PageSize_AutoDetect_MinPageWidthInPoints
	The minimum width of an auto-detected page size. The page will not be smaller than this width, but it can be wider. This is entered in points. There are 72 points per inch, so a default width of 576 is 8 inches.
Values:	String value of the minimum page width in points. The default is 576 , or 8 inches.

Conversion Settings -PNBuiltinTextConverter - Text With No Form Feeds Options

Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentNoFormFeeds.PageSize_AutoDetect_MinPageHeightInPoints
	The minimum height of an auto-detected page size. The page will not be shorter than this width, but it can be taller. There are 72 points per inch, so a height of 576 is 8 inches.
Values:	String value of the minimum page height in points. The default is 576 , or 8 inches.
Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentNoFormFeeds.PageSize_DefaultPageWidthInPoints
	Sets the desired page width in points. This setting is used if <i>PageSizeDetectionMode</i> is set to <i>UseSpecified</i> , or if it is set to <i>AutoDetect</i> and it does not match an appropriate page size. There are 72 points per inch, so the default width of 612 is 8.5 inches.
Values:	String value of the desired page width in points. The default is 612 , or 8.5 inches.
Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentNoFormFeeds.PageSize_DefaultPageHeightInPoints
	Sets the desired page height in points. This setting is used if <i>PageSizeDetectionMode</i> is set to <i>UseSpecified</i> , or if it is set to <i>AutoDetect</i> and it does not match an appropriate page size. There are 72 points per inch, so the default height of 792 is 11 inches.
Values:	String value of the desired page height in points. The default is 792 , or 11 inches.
Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentNoFormFeeds.PageSize_LeftMarginInPoints
	Sets the width of the left margin. A margin is the space between the edge of the page and the main body of text. There are 72 points per inch, so the default width of 18 is 0.25 inches.
Values:	String value of the desired left margin width in points. The default is 54 , or 0.75 inches.
Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentNoFormFeeds.PageSize_RightMarginInPoints
	Sets the width of the right margin. A margin is the space between the edge of the page and the main body of text. There are 72 points per inch, so the default width of 18 is 0.25 inches.
Values:	String value of the desired right margin width in points. The default is 54 , or 0.75 inches.
Name:	ConverterPlugIn.PNBuiltinTextConverterSettingsContentNoFormFeeds.PageSize_TopMarginInPoints
	Sets the height of the top margin. A margin is the space between the edge of the page and the main body of text. There are 72 points per inch, so the default height of 18 is 0.25 inches.
Values:	String value of the desired top margin height in points. The default is 72 , or 1 inch.

Conversion Settings -PNBuiltinTextConverter - Text With No Form Feeds Options

Name: **ConverterPlugIn.PNBuiltinTextConverterSettingsContentNoFormFeeds.PageSize_BottomMarginInPoints**

Sets the height of the bottom margin. A margin is the space between the edge of the page and the main body of text. There are 72 points per inch, so the default height of 18 is 0.25 inches.

Values: String value of the desired bottom margin height in points. The default is **72**, or 1 inch.

Name: **ConverterPlugIn.PNBuiltinTextConverterSettingsContentNoFormFeeds.TabStopEveryNSpaces**

Replaces any tab stops in the text document with the given number of spaces (columns).

Values: String value of the number of spaces to replace the tab stop with. The default is **8** spaces.

Built-in Cadd Converter Options

The Built-in Cadd Converter converts the following types of **Computer-Aided Design** and **Drafting** files: DWF, DWFX, PLT and GBX. It does not require any third-party applications.



Caution

This feature is not supported on Microsoft® Windows Server 2008 R2 and Microsoft® Windows 7.

Table values in **bold** text are the default value for that setting.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="TIFF 300dpi from CAD"
  Description ="Converts Cadd Files with optional password.">
  <Settings>

    <add Name ="ConverterPlugIn.PNBuiltinCaddConverterSettings.OpenPassword"
Value="password"/>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name="Save;Output File Format" Value="TIFF Multipaged"/>
    ...

  </Settings>
</Profile>
```

Conversion Settings -PNBuiltinCaddConverter

Name:	ConverterPlugIn.PNBuiltinCaddConverterSettings.OpenPassword
Values:	Applies to password protected DWF and DWFX files. Pass in the password to use to open the files. String value; the password to use to try an open the file.

Built-in Image Converter Options

The Built-in Image Converter is a replacement for our original [Image Converter](#). We've improved the speed and memory use needed when converting, modifying, and rotating images. This converter improves image conversion speed and provides better handling of all image types.

**Caution**

This feature is not supported on Microsoft® Windows Server 2008 R2 and Microsoft® Windows 7.

This converter has no individual options.

Word Converter Options

These options control the behavior of the Word converter used by Document Conversion Service. Table values in **bold** text are the default value for that setting.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="TIFF 300dpi Word with Markup, replace dates "
  Description ="Prints Word documents with comments and tracking visible.">
  <Settings>

    <!-- Print Word with markup and 2 pages per sheet -->
    <add Name ="Microsoft.Word.Document.PrintOut.Item"
      Value="DocumentAndMarkup"/>
    <add Name ="Microsoft.Word.PageSetup.TwoPagesOnOne"
      Value="True"/>

    <!-- Replace date fields with <AUTODATE> string -->
    <add Name ="Microsoft.Word.ReplaceFieldDateWith"
      Value="&lt;AUTODATE&gt;"/>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name="Save;Output File Format" Value="TIFF Multipaged"/>
    ...

  </Settings>
</Profile>
```

Conversion Settings - Word Printing Options	
Name:	Microsoft.Word.Document.PrintOut.Item Choose what parts of the document to print.
Values:	Document - prints only the document. DocumentAndMarkup - prints the document and any markup such as tracked changes and comments. DocumentMarkup - prints only the markup. DocumentProperties - prints only the document properties.
Name:	Microsoft.Word.Document.PrintOut.PageType Choose if you want to print all pages, even pages or odd pages.
Values:	All Even Odd
Name:	Microsoft.Word.ActiveWindow.View.MarkupMode Sets the display mode for tracked changes in the document. Applies when using the printing option <i>Word.Document.PrintOut.Item</i> set to <i>DocumentAndMarkup</i> or <i>DocumentMarkup</i> .
Values:	BalloonRevisions - Displays revisions in balloons in the left or right margin. InlineRevisions - Displays revisions within the text using strikethrough for deletions and underlining for insertions. MixedRevisions - Shows only comments and formatting revisions in the document.

Conversion Settings - Word Printing Options

Name: **Microsoft.Word.ActiveWindow.View.RevisionsView (Office 2010 and earlier)**

This setting is deprecated starting with Office 2013. Use **Microsoft.Word.ActiveWindow.View.RevisionsFilter.View** and **Microsoft.Word.ActiveWindow.View.RevisionsFilter.Markup** instead. Specifies whether the original version of a document or a version with revisions and formatting changes applied are displayed.

Values: **ViewFinal** - Displays the document with formatting and content changes applied.
ViewOriginal - Displays the document before changes were made.

Name: **Microsoft.Word.ActiveWindow.View.RevisionsFilter.View (Office 2013 and later)**

Specifies whether the original version of a document or a version with revisions and formatting changes applied are displayed. Replaces **Microsoft.Word.ActiveWindow.View.RevisionsView** in Office 2013 and later versions.

Values: **ViewFinal** - Displays the document with formatting and content changes applied.
ViewOriginal - Displays the document before changes were made.

Name: **Microsoft.Word.ActiveWindow.View.RevisionsFilter.Markup (Office 2013 and later)**

Specifies the extent of reviewer markup displayed in the document. This setting is used starting with Office 2013.

Values: **NoMarkup** - Displays the final document with no markup visible.
SimpleMarkup - Displays the final document in simple markup: with revisions incorporated, but with no markup visible.
AllMarkup - Displays the final document with all markup visible.

Name: **Microsoft.Word.ActiveWindow.View.ShowComments**

Pass *True* to display any comments in the document. Must be used with *Microsoft.Word.ActiveWindow.View.MarkupMode* to display the comments as balloons or inline, and *Microsoft.Word.Document.PrintOut.Item* set to print document markup.

Values: String value "True" or "False".

Name: **Microsoft.Word.ActiveWindow.View.ShowFormatChanges**

Pass *True* to display any formatting changes made to a document with Track Changes enabled. Must be used with *Microsoft.Word.ActiveWindow.View.MarkupMode* to display the comments as balloons or inline, and *Microsoft.Word.Document.PrintOut.Item* set to print document markup.

Values: String value "True" or "False".

Name: **Microsoft.Word.ActiveWindow.View.ShowHiddenText**

Pass *True* to display any text that was formatted as hidden.

Values: String value "True" or "False".

Conversion Settings - Word Printing Options

Name: **Microsoft.Word.ActiveWindow.View.ShowHighlight**

Pass *True* to have highlighted text displayed with the highlighted background.

Values: String value "True" or "False".

Name: **Microsoft.Word.ActiveWindow.View.ShowInkAnnotations**

Pass *True* to to show handwritten ink annotations in the document. Must be used with *Microsoft.Word.Document.PrintOut.Item* set to print document markup.

Values: String value "True" or "False".

Name: **Microsoft.Word.ActiveWindow.View.ShowInsertionsAndDeletions**

Pass *True* to display any insertions and deletions made to a document with Track Changes enabled. Must be used with *Microsoft.Word.ActiveWindow.View.MarkupMode* set to display the changes as balloons or inline, and *Microsoft.Word.Document.PrintOut.Item* set to print document markup.

Values: String value "True" or "False".

Name: **Microsoft.Word.ActiveWindow.View.ShowMarkupAreaHighlight**

Pass *True* to have the markup area that shows revision and comment balloons displayed shaded. Applies only when *Microsoft.Word.ActiveWindow.View.MarkupMode* is set to display markup as balloons, and *Microsoft.Word.Document.PrintOut.Item* is set to print document markup.

Values: String value "True" or "False".

Name: **Microsoft.Word.Options.AllowA4LetterResizing**

Pass *True* to automatically adjust Letter-sized documents to fit A4 paper, or to adjust A4-sized documents to fit Letter paper. This only affects printing and happens when the paper size of the printer does not match the paper size that is set in Word.

Values: String value "True" or "False".

Conversion Settings - Word Field Replacement

Name: **Microsoft.Word.ReplaceFieldDateWith**

Replaces any DATE fields in the Word document with the provided string.

Values: The string value to place in the field.

Name: **Microsoft.Word.ReplaceFieldTimeWith**

Replaces any TIME fields in the Word document with the provided string.

Values: The string value to place in the field.

Conversion Settings - Word Field Replacement

Name: **Microsoft.Word.ReplaceFieldFileNameWith**

Replaces any FILENAME fields in the Word document with the provided string.

Values: A string value to replace the auto file name field.

Conversion Settings - Word Document Protection

Name: **Microsoft.Word.UnprotectPassword**

The password to use to remove the protection on the the Word document and allow changes. This password is passed as clear text and is visible to anyone.

Values: A string value containing the password.

Name: **Microsoft.Word.OpenPassword**

The password to use to open a password-protected Word document. This password is passed as clear text and is visible to anyone.

Values: A string value containing the password.

Name: **Microsoft.Word.WritePassword**

The password to use to allow saving changes to the Word document. This password is passed as clear text and is visible to anyone.

Values: A string value containing the password.

Conversion Settings - Word Page Setup Printing Options

Name: **Microsoft.Word.PageSetup.BookFoldPrinting**

Pass *True* to print the document as a booklet.

Values: String value "True" or "False".

Name: **Microsoft.Word.PageSetup.BookFoldPrintingSheets**

The number pages to print in each booklet. This number must be a multiple of 4. If not, the default setting of "Auto" will be used.

When using "Auto", Word will automatically determine the number of sheets per booklet, splitting the sheets into separate booklets as necessary. Passing "All" will print all of your pages in a single booklet.

Values: String value "**Auto**", "All" or the number of pages to be printed in each booklet.

Conversion Settings - Word Page Setup Printing Options

Name:	Microsoft.Word.PageSetup.BookFoldRevPrinting
	Pass <i>True</i> to reverse the printing order for booklet printing, bidirectional or Asian language documents only.
Values:	String value "True" or "False".
Name:	Microsoft.Word.PageSetup.BottomMargin
	Set the size of the bottom margin in points.
Values:	String value of the desired margin height.
Name:	Microsoft.Word.PageSetup.DifferentFirstPageHeaderFooter
	Pass <i>True</i> to use a different header on the first page.
Values:	String value "True" or "False".
Name:	Microsoft.Word.PageSetup.FooterDistance
	Set the distance (in points) between the top of the footer to the bottom of the page.
Values:	String value of the desired footer height.
Name:	Microsoft.Word.PageSetup.Gutter
	Set the amount of extra margin space added for binding.
Values:	String value of the desired gutter width.
Name:	Microsoft.Word.PageSetup.GutterPos
	Sets which side of the document the gutter is placed.
Values:	Left Right Top
Name:	Microsoft.Word.PageSetup.GutterStyle
	Sets how the gutters are placed; on the left for left-to-right languages or on the right side of the document for right-to-left languages.
Values:	Bidi - use bidirectional gutters for right-to-left languages. Latin - use Latin gutter for left-to-right text.

Conversion Settings - Word Page Setup Printing Options

Name:	Microsoft.Word.PageSetup.HeaderDistance
	Set the distance (in points) between the bottom of the header to the top of the page.
Values:	String value of the desired header height.
Name:	Microsoft.Word.PageSetup.LayoutMode
	Sets the layout of the text in the document. Genko, Grid and LineGrid use the setting Microsoft.Word.PageSetup.LinesPage.
Values:	Default - No grid is used to lay out text. Genko - Text is laid out on a grid with characters aligned on the gridlines. Grid - Text is laid out on a grid but the characters are not aligned on the gridlines. LineGrid - Text is laid out on a grid; only the number of lines is specified.
Name:	Microsoft.Word.PageSetup.LeftMargin
	Set the size of the left margin in points.
Values:	String value of the desired margin height.
Name:	Microsoft.Word.PageSetup.LinesPage
	The number of lines per page of the document. Used with the Microsoft.Word.PageSetup.LayoutMode setting.
Values:	String value of the desired number of lines per page.
Name:	Microsoft.Word.PageSetup.MirrorMargins
	Pass <i>True</i> to have the inside and outside margins of facing pages to be the same width.
Values:	String value "True" or "False".
Name:	Microsoft.Word.PageSetup.OddAndEvenPagesHeaderFooter
	Pass <i>True</i> to have different headers for odd-numbered and even-numbered pages.
Values:	String value "True" or "False".
Name:	Microsoft.Word.PageSetup.Orientation
	Sets the orientation of the page.
Values:	Landscape Portrait

Conversion Settings - Word Page Setup Printing Options**Name:** **Microsoft.Word.PageSetup.PageHeight**

Sets the height of the page in points.

Values: String value of the desired height.**Name:** **Microsoft.Word.PageSetup.PageWidth**

Sets the width of the page in points.

Values: String value of the desired width.**Name:** **Microsoft.Word.PageSetup.PaperSize**

Sets the paper size.

Values:

- Paper10x14 - 10 in. x 14 in.
- Paper11x17 - 11 in. x 17 in.
- PaperA3 - A3 (297 mm x 420 mm)
- PaperA4 - A4 (210 mm x 297 mm)
- PaperA4Small - A4 Small (210 mm x 297 mm)
- PaperA5 - A5 (148 mm x 210 mm)
- PaperB4 - B4 (250 mm x 354 mm)
- PaperB5 - B5 (182 mm x 257 mm)
- PaperCsheet - C size sheet
- PaperEnvelope10 - Envelope #10 (4-1/8 in. x 9-1/2 in.)
- PaperEnvelope11 - Envelope #11 (4-1/2 in. x 10-3/8 in.)
- PaperEnvelope14 - Envelope #14 (5 in. x 11-1/2 in.)
- PaperEnvelope9 - Envelope #9 (3-7/8 in. x 8-7/8 in.)
- PaperEnvelopeB4 - Envelope B4 (250 mm x 353 mm)
- PaperEnvelopeB5 - Envelope B5 (176 mm x 250 mm)
- PaperEnvelopeB6 - Envelope B6 (176 mm x 125 mm)
- PaperEnvelopeC3 - Envelope C3 (324 mm x 458 mm)
- PaperEnvelopeC4 - Envelope C4 (229 mm x 324 mm)
- PaperEnvelopeC5 - Envelope C5 (162 mm x 229 mm)
- PaperEnvelopeC6 - Envelope C6 (114 mm x 162 mm)
- PaperEnvelopeC65 - Envelope C65 (114 mm x 229 mm)
- PaperEnvelopeDL - Envelope DL (110 mm x 220 mm)
- PaperEnvelopeItaly - Envelope (110 mm x 230 mm)
- PaperEnvelopeMonarch - Envelope Monarch (3-7/8 in. x 7-1/2 in.)
- PaperEnvelopePersonal - Envelope (3-5/8 in. x 6-1/2 in.)
- PaperExecutive - Executive (7-1/2 in. x 10-1/2 in.)
- PaperFanfoldLegalGerman - German Legal Fanfold (8-1/2 in. x 13 in.)
- PaperFanfoldStdGerman - German Standard Fanfold (8-1/2 in. x 12 in.)
- PaperFolio - Folio (8-1/2 in. x 13 in.)
- PaperLedger - Ledger (17 in. x 11 in.)
- PaperLegal - Legal (8-1/2 in. x 14 in.)
- PaperLetter - Letter (8-1/2 in. x 11 in.)
- PaperLetterSmall - Letter Small (8-1/2 in. x 11 in.)
- PaperNote - Note (8-1/2 in. x 11 in.)
- PaperQuarto - Quarto (215 mm x 275 mm)
- PaperStatement - Statement (5-1/2 in. x 8-1/2 in.)
- PaperTabloid - Tabloid (11 in. x 17 in.)

Conversion Settings - Word Page Setup Printing Options

Name: **Microsoft.Word.PageSetup.RightMargin**

Set the size of the right margin in points.

Values: String value of the desired margin width.

Name: **Microsoft.Word.PageSetup.SuppressEndnotes**

Pass *True* to suppress any endnotes.

Values: String value "True" or "False".

Name: **Microsoft.Word.PageSetup.TopMargin**

Set the size of the top margin in points.

Values: String value of the desired margin height.

Name: **Microsoft.Word.PageSetup.TwoPagesOnOne**

Pass *True* to split the paper right down the horizontal center (for portrait) and vertical center (for landscape) and print two "pages" per sheet of paper. This does not shrink two pages of the document onto each single output page but rather changes the text layout of the document to reflect each page size being one half of the currently selected paper size.

Values: String value "True" or "False".

Name: **Microsoft.Word.PageSetup.VerticalAlignment**

Sets the vertical alignment of the text on each page.

Values: Bottom
Center
Justify
Top

Excel Converter Options

These options control the behavior of the Excel converter used by Document Conversion Service. If the workbook, or any spreadsheet in the workbook is password protected and the password is not known, the options are ignored. The settings cannot be applied to a protected workbook or spreadsheet.

Table values in **bold** text are the default value for that setting. Not all settings have default values; these settings are optional and the appropriate setting in the spreadsheet being printed will be used.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="TIFF 300dpi Excel Charts First, replace dates "
  Description ="Prints Excel charts, sheets with grid lines.">
  <Settings>

    <!-- Print Charts then workbooks, show grid lines -->
    <add Name ="Microsoft.Excel.PrintOut" Value="PrintOutChartsThenWorkbook"/>
    <add Name ="Microsoft.Excel.PageSetup.PrintGridlines" Value="True"/>

    <!-- Replace header/footer date fields with <AUTODATE> string -->
    <add Name ="Microsoft.Excel.ReplaceFieldDateWith"
      Value="&lt;AUTODATE&gt;"/>
    <add Name ="Microsoft.Excel.PageSetup.LeftHeader"
      Value="Sheet: &A"/>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name="Save;Output File Format" Value="TIFF Multipaged"/>
    ...

  </Settings>
</Profile>
```

Conversion Settings - Excel General Formatting & Printing Options

Name: **Microsoft.Excel.PrintOut**

Choose what part of the Excel spreadsheet to print.
The settings

For *PrintOutChartsOnly*, *PrintOutChartsThenWorkbook* and *PrintOutWorkbookThenCharts*, the option *Microsoft.Excel.PrintOut.PrintEmbeddedChartsFirst* controls if embedded charts are printed before or after any chart tabs in the spreadsheet.

Values: **PrintOutWorkbookOnly** - prints the entire workbook just as Excel does.

PrintOutActiveSheetOnly - prints only the last active (selected) sheet in the workbook. This is the selected tab at the time the Excel file was last saved.

PrintOutSelectedSheetsOnly - prints only the selected sheets in the workbook. Multiple sheets can be selected using the Ctrl+Left Click with the mouse.

PrintOutSheetsWithPrintAreasOnly - prints only sheets that have a print area set.

PrintOutChartsOnly - prints any charts tabs and embedded charts in the workbook.

PrintOutChartsThenWorkbook - prints all chart tabs and embedded charts, then prints all sheets in the workbook.

PrintOutWorkbookThenCharts - prints all sheets in the workbook, then prints all chart tabs and embedded charts.

For the three options above, embedded charts can be before or after other charts, as specified by the *Microsoft.Excel.PrintOut.PrintEmbeddedChartsFirst* setting.

Name: **Microsoft.Excel.PrintHiddenWorksheets**

Choose whether to print hidden worksheets or not.

Values: **False** - do not print hidden worksheets.
True - print hidden worksheets.**Name:** **Microsoft.Excel.PrintOut.PrintEmbeddedChartsFirst**

When printing embedded charts, determines if the embedded charts are printed before or after any chart tabs in the spreadsheet. Applies only when *Microsoft.Excel.PrintOut* is set to print charts.

Values: **False** - print embedded charts after all other charts.
True - print embedded charts first.

Conversion Settings - Excel General Formatting & Printing Options

Name: **Microsoft.Excel.PrintSheetsRangeByIndex**

The sheet numbers and ranges to include when printing. Separate each number and range with a comma. For example, "1, 3-5" prints sheet 1 and sheets 3 through 5. Numbers in the range exceeding the sheet count of the source document are ignored.

Sheet numbers in the range are for visible sheets unless *Microsoft.Excel.PrintHiddenWorksheets* is true, then hidden sheets are included.

Applies to the **Microsoft.Excel.PrintOut** options *PrintOutWorkbookOnly*, *PrintOutChartsOnly*, *PrintOutChartsThenWorkbook* and *PrintOutWorkbookThenCharts*. The range applies to both sheets and charts in the workbook.

This print filter can be combined with *Microsoft.Excel.PrintSheetsRangeByName*, *Microsoft.Excel.PrintFirstNSheets*, *Microsoft.Excel.PrintLastNSheets*, and *Microsoft.Excel.PrintIfSheetNameMatchesRegex*.

Values: The string representing the numbered sheet range.

Name: **Microsoft.Excel.PrintSheetsRangeByName**

The names of the sheets and charts to include when printing, separated with a colon symbol (:) to print multiple sheets. Names not in the worksheet collection are ignored.

Applies only to visible sheets unless *Microsoft.Excel.PrintHiddenWorksheets* is true.

Applies to the **Microsoft.Excel.PrintOut** options *PrintOutWorkbookOnly*, *PrintOutChartsOnly*, *PrintOutChartsThenWorkbook* and *PrintOutWorkbookThenCharts*. The name selection applies to both sheets and charts in the workbook.

This print filter can be combined with *Microsoft.Excel.PrintSheetsRangeByIndex*, *Microsoft.Excel.PrintFirstNSheets*, *Microsoft.Excel.PrintLastNSheets* and *Microsoft.Excel.PrintIfSheetNameMatchesRegex*.

Values: The string of sheet or chart names, such as "Sheet1:Sheet3:Chart1".

Name: **Microsoft.Excel.PrintFirstNSheets**

Includes the designated number of sheets or charts, starting at the beginning of the workbook. If the workbook has less sheets (tabs) in total than the requested number, all sheets are printed.

Applies only to visible sheets unless *Microsoft.Excel.PrintHiddenWorksheets* is true.

Applies to the **Microsoft.Excel.PrintOut** options *PrintOutWorkbookOnly*, *PrintOutChartsOnly*, *PrintOutChartsThenWorkbook* and *PrintOutWorkbookThenCharts*. Applies to both sheets and charts in the workbook.

This print filter can be combined with *Microsoft.Excel.PrintSheetsRangeByName*, *Microsoft.Excel.PrintSheetsRangeByIndex*, *Microsoft.Excel.PrintLastNSheets*, and *Microsoft.Excel.PrintIfSheetNameMatchesRegex*.

Values: The number of sheets to print.

Conversion Settings - Excel General Formatting & Printing Options

Name: **Microsoft.Excel.PrintLastNSheets**

Includes the last designated number of sheets or charts, starting in the middle and going to the end of the workbook. If the workbook has less sheets (tabs) in total than the requested number, all sheets are printed.

Applies only to visible sheets unless *Microsoft.Excel.PrintHiddenWorksheets* is true.

Applies to the **Microsoft.Excel.PrintOut** options *PrintOutWorkbookOnly*, *PrintOutChartsOnly*, *PrintOutChartsThenWorkbook* and *PrintOutWorkbookThenCharts*. Applies to both sheets and charts in the workbook.

This print filter can be combined with *Microsoft.Excel.PrintSheetsRangeByName*, *Microsoft.Excel.PrintSheetsRangeByIndex*, *Microsoft.Excel.PrintFirstNSheets* and *Microsoft.Excel.PrintIfSheetNameMatchesRegex*.

Values: The number of sheets to print.

Name: **Microsoft.Excel.PrintIfSheetNameMatchesRegex**

Includes the sheet or chart if its name matches the regular expression.

Applies only to visible sheets unless *Microsoft.Excel.PrintHiddenWorksheets* is true.

Applies to the **Microsoft.Excel.PrintOut** options *PrintOutWorkbookOnly*, *PrintOutChartsOnly*, *PrintOutChartsThenWorkbook* and *PrintOutWorkbookThenCharts*. Applies to both sheets and charts in the workbook.

This print filter can be combined with *Microsoft.Excel.PrintSheetsRangeByIndex*, *Microsoft.Excel.PrintSheetsRangeByName*, *Microsoft.Excel.PrintFirstNSheets* and *Microsoft.Excel.PrintLastNSheets*.

Values: The regular expression to match the sheet name against.

Name: **Microsoft.Excel.AutoFit.KeepEmbeddedChartScaling**

Applies only when *Microsoft.Excel.AutoFitRows* and *Microsoft.Excel.AutoFitColumns* are set and if one or more embedded charts are on the sheet. When *True*, the width and height of any rows and columns under embedded charts are not auto-adjusted so that the chart does not change shape. Default is *True*.

Values: False - autofit all rows or columns, even under embedded charts. This can cause any charts to be squished or stretched.
True - do not autofit rows and columns under embedded charts; charts will keep their original scaling on the sheet.

Name: **Microsoft.Excel.Worksheet.IncludeCellFormulasAsComments**

For any cell that contains a formula, the formula added to that cell as a comment. If the cell has a comment, the formula is inserted with a carriage return before any current comment text. This must be used with *Microsoft.Excel.PageSetup.PrintComments* set to *PrintSheetEnd* to include the cell formulas listed by cell reference at the end of each sheet.

To append the formula to the cell contents instead of inserting at the beginning, set *Microsoft.Excel.Worksheet.PrependCellFormulaToCommentText* to False.

Values: False - do not add/update existing comments with the cell formula.
True - add/update existing comment with the cell formula.

Conversion Settings - Excel General Formatting & Printing Options

Name:	Microsoft.Excel.Worksheet.PrependCellFormulaToCommentText
	When using <i>Microsoft.Excel.Worksheet.IncludeCellFormulasAsComments</i> , the formula is prepended to the beginning of any existing comment text by default. To append the formula to the end of any existing comment text, set this option to <i>False</i> .
Values:	False - append the cell formula to the end of any existing comment text. True - prepend the cell formula to the beginning of any existing comment text.
Name:	Microsoft.Excel.Worksheet.PrintOut.IgnorePrintAreas
	When set to <i>True</i> , any print areas set on the worksheet will be ignored and the entire worksheet printed. Use with <i>Microsoft.Excel.Worksheet.PrintOut.ResetAllPageBreaks</i> to print the worksheet differently from the printing options in the worksheet.
Values:	False - prints using any print area set on the worksheet. True - prints the entire worksheet.
Name:	Microsoft.Excel.Worksheet.ShowAllData
	Makes all rows of any filtered data visible. This setting only applies to filtered data in the worksheet. To show hidden columns or rows use <i>Microsoft.Excel.AutoFitRows</i> and <i>Microsoft.Excel.AutoFitColumns</i> .
Values:	False - Leave data filtered (hidden). True - Show all the data on the worksheet.
Name:	Microsoft.Excel.Worksheet.ResetAllPageBreaks
	Set as <i>True</i> to resets all page breaks on each worksheet. Use with <i>Microsoft.Excel.Worksheet.PrintOut.IgnorePrintAreas</i> to print the worksheet differently from the printing options in the worksheet.
Values:	False - Leave page breaks alone. True - Reset all page breaks.
Name:	Microsoft.Excel.AutoFitRows
	If set to <i>True</i> then the height of the rows in the spreadsheet will be adjusted automatically to fit the contents of the cells. This setting will allow you to show all hidden rows in the worksheet.
Values:	String value "True" or "False".
Name:	Microsoft.Excel.AutoFitRows.Adjust
	This setting is only applied when <i>Microsoft.Excel.AutoFitRows</i> is set to "True" and allows you to add the height specified (in points) to all rows after they have been auto-fit. The maximum row height allowed in Excel is 409 points. It is not normally needed to add height to each row and adding height to each row can be a time-consuming operation; only use this option if absolutely needed.
Values:	String value of the amount, in points, by which to adjust the row height.

Conversion Settings - Excel General Formatting & Printing Options

Name: **Microsoft.Excel.AutoFitColumns**

If set to *True* then the width the columns in the spreadsheet will be adjusted to fit the contents of the cells. This setting will allow you to show all hidden columns in the worksheet.

Values: String value "True" or "**False**".

Name: **Microsoft.Excel.AutoFitColumns.Adjust**

This setting is only applied when *Microsoft.Excel.AutoFitColumns* is set to "True" and allows you to add the width specified (in points) to all columns after they have been auto-fit. The maximum column width allowed in Excel is 255 points. It is not normally needed to add width to each column and adding width to each column can be a time-consuming operation; only use this option if absolutely needed.

Values: String value of the amount, in points, by which to adjust the column width.

Name: **Microsoft.Excel.AutoFit.KeepEmbeddedChartScaling**

Only applies when auto-fit rows and columns is enabled. When set to its default of "True", autofit is not applied to any rows and/or columns that are under any embedded charts on the sheet. All other rows and columns are auto-fit. This allows the embedded charts to maintain the scale they were originally set at when placed on the spreadsheet. If set to "False", the chart will change size depending on the new height and width of the underlying rows and columns.

Values: String value "**True**" or "False".

Name: **Microsoft.Excel.UnfreezePanels**

If the spreadsheet has any non-scrolling, "frozen" panes, pass "True" to unfreeze them before printing.

Values: String value "True" or "**False**".

Name: **Microsoft.Excel.ClearFormatsOnEmptyRowsOnTop**

Clears the formatting of any empty rows (cells with no data) at the top of the spreadsheet so that only rows with data in them are printed.

Values: String value "True" or "**False**".

Name: **Microsoft.Excel.ClearFormatsOnEmptyRowsOnBottom**

Clears the formatting of any empty rows (cells with no data) at the bottom of the spreadsheet so that only rows with data in them are printed.

Values: String value "True" or "**False**".

Name: **Microsoft.Excel.ClearFormatsOnEmptyColumnsOnLeft**

Clears the formatting of any empty columns (cells with no data) on the left hand side of the spreadsheet so that only columns with data in them are printed.

Values: String value "True" or "**False**".

Conversion Settings - Excel General Formatting & Printing Options

Name: **Microsoft.Excel.ClearFormatsOnEmptyColumnsOnRight**

Clears the formatting of any empty columns (cells with no data) on the right hand side of the spreadsheet so that only columns with data in them are printed.

Values: String value "True" or "**False**".

Name: **Microsoft.Excel.RemoveBackgroundColors**

Clears the background colors and fills for all cells. Leaves text color and borders unchanged.

Note: This does not apply to cells that have conditional formatting applied.

Values: String value "True" or "**False**".

Name: **Microsoft.Excel.SetAllTextAsBlack**

Sets all text to black.

Note: This does not apply to cells that have conditional formatting applied.

Values: String value "True" or "**False**".

Name: **Microsoft.Excel.ClearTableStyle**

Clears the table styling from any columns or rows in the spreadsheet. Leaves the cell data, formatting and formulas in place. This can be a time-consuming operation as the table formatting is copied to each cell; only use this option if absolutely needed. To do the same but also remove the formatting, use *Microsoft.Excel.ClearTableStyleAndFormatting*.

Values: String value "True" or "**False**".

Name: **Microsoft.Excel.ClearTableStyleAndFormatting**

Clears the table styling and any table formatting from any columns or rows in the spreadsheet. Leaves the cell data and formulas in place.

Values: String value "True" or "**False**".

Name: **Microsoft.Excel.ClearAllConditionalFormatting**

Clears all conditional formatting applied to any cells. This includes removing background colors and text styling, color scales, data bars and icon sets.

Note: This does not apply to any spreadsheet that is protected or shared.

Values: String value "True" or "**False**".

Conversion Settings - Excel General Formatting & Printing Options

Name: **Microsoft.Excel.TrackChanges.HighlightChangesOnScreen**

If *Track Changes* has been enabled for the workbook, any cell on any spreadsheet that has been changed will be highlighted.

Values: String value "True" or "**False**".

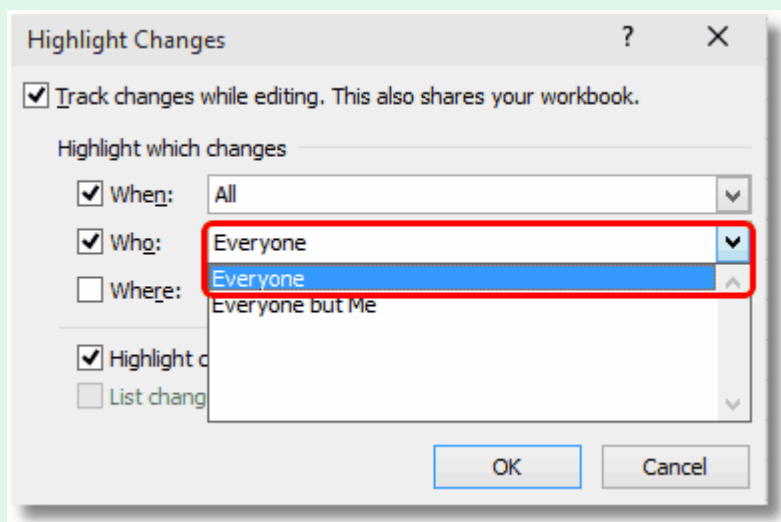
Name: **Microsoft.Excel.TrackChanges.ListChangesOnNewSheet**

If *Track Changes* has been enabled for the workbook, setting this to *True* will create a new temporary, protected spreadsheet that lists all of changes made to the workbook. If not using the English version of Excel, *Microsoft.Excel.TrackChanges.ExcelTrackChangesWhoParameter* will also need to be set.

Values: String value "True" or "**False**".

Name: **Microsoft.Excel.TrackChanges.ExcelTrackChangesWhoParameter**

When using an Office installation in a language other than English, this option must specify the word "Everyone" in that language to list the tracked changes for all users. The default for this setting is "Everyone". The 5 most common languages are listed below, or you can find the needed parameter on the Highlight Changes dialog in your version of Excel. The English version is shown below.



Values: English - **Everyone**
 French - Tous, Tout le monde
 Italian - Tutti
 German - Jeder
 Spanish - Todos

Conversion Settings - Excel Page Setup Printing Options

Name: **Microsoft.Excel.PageSetup.AlignMarginsHeaderFooter**

Have Excel align the header and the footer with the margins set in the page setup options.

Values: String value "True" or "**False**".

Conversion Settings - Excel Page Setup Printing Options

Name:	Microsoft.Excel.PageSetup.BlackAndWhite Print the Excel document in black and white.
Values:	String value "True" or "False".
Name:	Microsoft.Excel.PageSetup.BottomMargin Set the size of the bottom margin in points.
Values:	String value of the desired margin height.
Name:	Microsoft.Excel.PageSetup.CenterFooter The text to display in the center footer area of the worksheet.
Values:	String value of the text to display.
Name:	Microsoft.Excel.PageSetup.CenterHeader The text to display in the center header area of the worksheet.
Values:	String value of the text to display.
Name:	Microsoft.Excel.PageSetup.CenterHorizontally Center the worksheet horizontally on the page when printed.
Values:	String value "True" or "False".
Name:	Microsoft.Excel.PageSetup.CenterVertically Center the worksheet vertically on the page when printed.
Values:	String value "True" or "False".
Name:	Microsoft.Excel.PageSetup.DifferentFirstPageHeaderFooter If this is <i>True</i> a different header or footer is used for the first page of the worksheet (<i>applies to Office 2007 or higher</i>).
Values:	String value "True" or "False".
Name:	Microsoft.Excel.PageSetup.Draft Prints the worksheet without graphics when set to <i>True</i> .
Values:	String value "True" or "False".

Conversion Settings - Excel Page Setup Printing Options

Name: **Microsoft.Excel.PageSetup.FirstPageNumber**

Sets the first page number that will be used when this sheet is printed.

Values: String value of the page number to start with.

Name: **Microsoft.Excel.PageSetup.FitToPagesTall**

Set the number of pages tall the worksheet will scale to when printed. Ignored when Microsoft.Excel.PageSetup.Zoom is set to *True*.

Values: String value of the number of pages tall to use or "False" to use the scaling set in the Microsoft.Excel.PageSetup.FitToPagesWide setting.

Name: **Microsoft.Excel.PageSetup.FitToPagesWide**

Set the number of pages wide the worksheet will scale to when printed. Ignored when Microsoft.Excel.PageSetup.Zoom is set to *True*.

Values: String value of the number of pages wide to use or "False" to use the scaling set in the Microsoft.Excel.PageSetup.FitToPagesTall setting.

Name: **Microsoft.Excel.PageSetup.FooterMargin**

Sets the distance, in points, from the bottom of the page to the footer.

Values: String value of the desired margin height.

Name: **Microsoft.Excel.PageSetup.HeaderMargin**

Sets the distance, in points, from the top of the page to the header.

Values: String value of the desired margin height.

Name: **Microsoft.Excel.PageSetup.LeftFooter**

The text to display in the left footer area of the worksheet.

Values: String value of the text to display.

Name: **Microsoft.Excel.PageSetup.LeftHeader**

The text to display in the left header area of the worksheet.

Values: String value of the text to display.

Name: **Microsoft.Excel.PageSetup.LeftMargin**

Set the size of the left margin in points.

Values: String value of the desired margin height.

Conversion Settings - Excel Page Setup Printing Options

Name:	Microsoft.Excel.PageSetup.OddAndEvenPagesHeaderFooter
	Set to <i>True</i> if different headers and footers have been set for odd-numbered and even-numbered pages.
Values:	String value "True" or "False".
Name:	Microsoft.Excel.PageSetup.Order
	Choose the page order when printing multiple spreadsheet pages per page.
Values:	DownThenOver - print the spreadsheet pages down then across the page. OverThenDown - print the spreadsheet pages across the page, then down.
Name:	Microsoft.Excel.PageSetup.Orientation
	Choose the orientation of the Excel spreadsheet.
Values:	Landscape Portrait

Conversion Settings - Excel Page Setup Printing Options

Name: **Microsoft.Excel.PageSetup.PaperSize**

Sets the size of the paper the worksheet will be printed on.

Values:

- Paper10x14 - 10 in. x 14 in.
- Paper11x17 - 11 in. x 17 in.
- PaperA3 - A3 (297 mm x 420 mm)
- PaperA4 - A4 (210 mm x 297 mm)
- PaperA4Small - A4 Small (210 mm x 297 mm)
- PaperA5 - A5 (148 mm x 210 mm)
- PaperB4 - B4 (257 mm x 364 mm)
- PaperB5 - B5 (182 mm x 257 mm)
- PaperCsheet - C size sheet
- PaperDsheet - D size sheet
- PaperEnvelope10 - Envelope #10 (4-1/8 in. x 9-1/2 in.)
- PaperEnvelope11 - Envelope #11 (4-1/2 in. x 10-3/8 in.)
- PaperEnvelope12 - Envelope #12 (4-1/2 in. x 11 in.)
- PaperEnvelope14 - Envelope #14 (5 in. x 11-1/2 in.)
- PaperEnvelope9 - Envelope #9 (3-7/8 in. x 8-7/8 in.)
- PaperEnvelopeB4 - Envelope B4 (250 mm x 353 mm)
- PaperEnvelopeB5 - Envelope B5 (176 mm x 250 mm)
- PaperEnvelopeB6 - Envelope B6 (176 mm x 125 mm)
- PaperEnvelopeC3 - Envelope C3 (324 mm x 458 mm)
- PaperEnvelopeC4 - Envelope C4 (229 mm x 324 mm)
- PaperEnvelopeC5 - Envelope C5 (162 mm x 229 mm)
- PaperEnvelopeC6 - Envelope C6 (114 mm x 162 mm)
- PaperEnvelopeC65 - Envelope C65 (114 mm x 229 mm)
- PaperEnvelopeDL - Envelope DL (110 mm x 220 mm)
- PaperEnvelopeItaly - Envelope (110 mm x 230 mm)
- PaperEnvelopeMonarch - Envelope Monarch (3-7/8 in. x 7-1/2 in.)
- PaperEnvelopePersonal - Envelope (3-5/8 in. x 6-1/2 in.)
- PaperEsheet - E size sheet
- PaperExecutive - Executive (7-1/2 in. x 10-1/2 in.)
- PaperFanfoldLegalGerman - German Legal Fanfold (8-1/2 in. x 12 in.)
- PaperFanfoldStdGerman - German Legal Fanfold (8-1/2 in. x 13 in.)
- PaperFanfoldUS - U.S. Standard Fanfold (14-7/8 in. x 11 in.)
- PaperFolio - Folio (8-1/2 in. x 13 in.)
- PaperLedger - Ledger (17 in. x 11 in.)
- PaperLegal - Legal (8-1/2 in. x 14 in.)
- PaperLetter - Letter (8-1/2 in. x 11 in.)
- PaperLetterSmall - Letter Small (8-1/2 in. x 11 in.)
- PaperNote - Note (8-1/2 in. x 11 in.)
- PaperQuarto - Quarto (215 mm x 275 mm)
- PaperStatement - Statement (5-1/2 in. x 8-1/2 in.)
- PaperTabloid - Tabloid (11 in. x 17 in.)

Name: **Microsoft.Excel.PageSetup.PrintArea**

Sets the range to be printed, as a string using Excel's A1-style references.

Values: String containing the print area. Pass an empty string to print the entire worksheet.

Conversion Settings - Excel Page Setup Printing Options

Name: **Microsoft.Excel.PageSetup.PrintComments**

Determines where any comments in the worksheet are printed.

Values: PrintSheetEnd - print the comments as notes at the end of the worksheet.
PrintInPlace - comments are printed in-place in the worksheet as pop-up notes.
PrintNoComments - comments are not printed.

Name: **Microsoft.Excel.PageSetup.PrintErrors**

Set the type of print error displayed.

Values: PrintErrorsDisplayed - display all print errors.
PrintErrorsBlank - print errors are blank.
PrintErrorsDash - display print errors as dashes.
PrintErrorsNA - display print errors as not available.

Name: **Microsoft.Excel.PageSetup.PrintGridlines**

If set to *True* then grid lines will be printed on each spreadsheet.

Values: String value "True" or "False".

Name: **Microsoft.Excel.PageSetup.PrintHeadings**

If set to *True* then column and row headings will be printed on each spreadsheet.

Values: String value "True" or "False".

Name: **Microsoft.Excel.PageSetup.PrintNotes**

Set to *True* to print cell notes as end notes with the worksheet.

Values: String value "True" or "False".

Name: **Microsoft.Excel.PageSetup.PrintQuality**

Sets the print quality, or DPI, of the worksheet. This is different from the *DevMode settings;Resolution* setting in the [Devmode settings](#) section.

Values: 1200, 720, 600, 400, 360, 300, 240, 200, 150, 120, 100, 75, 60, 50

Name: **Microsoft.Excel.PageSetup.PrintTitleColumns**

Sets the columns that contain the cells to be repeated on the left side of each page as a string using Excel's A1-style references.

Values: String containing the columns to use as title columns. Pass an empty string to turn off title columns.

Conversion Settings - Excel Page Setup Printing Options

Name: **Microsoft.Excel.PageSetup.PrintTitleRows**

Sets the rows that contain the cells to be repeated on the top of each page as a string using Excel's A1-style references.

Values: String containing the rows use as title rows. Pass an empty string to turn off title rows.

Name: **Microsoft.Excel.PageSetup.RightFooter**

The text to display in the right footer area of the worksheet.

Values: String value of the text to display.

Name: **Microsoft.Excel.PageSetup.RightHeader**

The text to display in the right header area of the worksheet.

Values: String value of the text to display.

Name: **Microsoft.Excel.PageSetup.RightMargin**

Set the size of the left margin in points.

Values: String value of the desired margin width.

Name: **Microsoft.Excel.PageSetup.ScaleWithDocHeaderFooter**

If set to *True* then the header and footer will be scaled with the document when the size of the document changes.

Values: String value "True" or "False".

Name: **Microsoft.Excel.PageSetup.TopMargin**

Set the size of the top margin in points.

Values: String value of the desired margin height.

Name: **Microsoft.Excel.PageSetup.Zoom**

Sets a percentage (between 10 and 400 percent) by which the worksheet will be scaled when printed.

Values: String value representing the zoom percentage, or "False" to use the `Microsoft.Excel.PageSetup.FitToPagesTall` and `Microsoft.Excel.PageSetup.FitToPagesWide` properties instead.

Conversion Settings - Excel Field Replacement**Name:** **Microsoft.Excel.ReplaceFieldDateWith**

Replaces any DATE fields in the header and footer in the Excel document with the provided string.

Values: The string value to place in the field.**Name:** **Microsoft.Excel.ReplaceFieldTimeWith**

Replaces any TIME fields in the header and footer in the Excel document with the provided string.

Values: The string value to place in the field.**Name:** **Microsoft.Excel.ReplaceFieldFileNameWith**

Replaces any FILENAME fields in the header and footer in the Excel document with the provided string.

Values: A string value to replace the auto file name field.**Name:** **Microsoft.Excel.ReplaceFormulasWithAutoDateAndTimeAsString**Replaces any cells containing a formula with the functions TODAY() and NOW() with the provided string. This will replace the **entire** cell formula.**Values:** A string value to display as the cell contents.**Conversion Settings - Document Protection****Name:** **Microsoft.Excel.UnprotectPassword**

The password is used to unprotect the Excel document and allow changes. This password is passed as clear text and is visible to anyone.

Values: A string value containing the password.**Name:** **Microsoft.Excel.OpenPassword**

The password is used to open a password-protected Excel document. This password is passed as clear text and is visible to anyone.

Values: A string value containing the password.**Name:** **Microsoft.Excel.WritePassword**

The password is used to allow saving changes to the Excel document. This password is passed as clear text and is visible to anyone.

Values: A string value containing the password.

Conversion Settings - Document Protection

Name: **Microsoft.Excel.RemoveDocumentProtection**

Does not apply to Excel 2013 and later versions.

Temporarily remove any workbook or spreadsheet protection that may be set on the document. This allows Excel printing and formatting options to be applied.

Values: String value "True" or "False". Default is True for Excel 2010 and previous versions. Ignored for Office 2013 and later.

Name: **Microsoft.Excel.SkipFileValidation**

Office File Validation is a security feature added starting with Microsoft Office 2010. This feature checks Office files created with older versions to ensure they were safe to open before actually opening them. Files can be marked as invalid if they are corrupt or contain malicious code. Unfortunately, this can also mean that files created previous versions of Office can mistakenly be tagged as invalid when they are not.

You can use this setting to disable this feature.

We do not recommend enabling this feature; you do so at your own risk. Use with caution and only disable if you know and trust the source of the Excel files.

Values: **False** - Files are always validated upon opening.
True - Skip file validation upon opening. ***This setting is not recommended.***

Header and Footer Formatting Codes

The following formatting codes are used to customize the header and footer contents of the spreadsheet with page numbers, the date, the name of the sheet, or the name and path of the file taken from the Excel file being converted.

Applies to these settings:

- Microsoft.Excel.PageSetup.LeftHeader
- Microsoft.Excel.PageSetup.CenterHeader
- Microsoft.Excel.PageSetup.RightHeader
- Microsoft.Excel.PageSetup.LeftFooter
- Microsoft.Excel.PageSetup.CenterFooter
- Microsoft.Excel.PageSetup.RightFooter

These formatting codes are applied to the header and footer contents **after** any auto date, time or filename replacement is applied from the settings *Microsoft.Excel.ReplaceFieldDateWith*, *Microsoft.Excel.ReplaceFieldTimeWith*, and *Microsoft.Excel.ReplaceFieldFileNameWith*.

This means that if you use an autodate, autotime or file name formatting code in a custom header, you will get the autodate, autotime or file name in the header or footer, and not the replacement string.

&P	Current page number
&N	Number of pages

&D	Auto date
&T	Auto time
&Z&F	Path to file
&F	File name
&A	Sheet name

PowerPoint Converter Options

These options control the behavior of the PowerPoint converter used by Document Conversion Service. Table values in **bold** text are the default value for that setting. Not all settings have default values; these settings are optional and the appropriate setting in the presentation being printed will be used.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="TIFF 300dpi Powerpoint Markup, replace dates "
  Description="Prints Powerpoint slides with comments and tracking visible.">
  <Settings>

    <add Name="Microsoft.PowerPoint.PageSetup.FirstSlideNumber" Value="2"/>
    <add Name="Microsoft.PowerPoint.PageSetup.NotesOrientation"
      Value="OrientationVertical"/>
    <add Name="Microsoft.PowerPoint.PrintOptions.FitToPage" Value="True"/>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name="Save;Output File Format" Value="TIFF Multipaged"/>
    ...

  </Settings>
</Profile>
```

Conversion Settings - PowerPoint Page Setup

Name:	Microsoft.PowerPoint.PageSetup.FirstSlideNumber
	Sets the slide number for the first slide in the presentation.
Values:	String value containing the starting number, such as "2".
Name:	Microsoft.PowerPoint.PageSetup.NotesOrientation
	Sets the printed orientation of notes pages, handouts, and outlines for the specified presentation. If the value passed down does not match the strings below, the orientation will default to OrientationHorizontal.
Values:	OrientationHorizontal OrientationVertical OrientationMixed
Name:	Microsoft.PowerPoint.PageSetup.SlideOrientation
	Sets the printed orientation of slides in the presentation. If the value passed down does not match the strings below, the orientation will default to OrientationHorizontal.
Values:	OrientationHorizontal OrientationVertical OrientationMixed
Name:	Microsoft.PowerPoint.PageSetup.SlideHeight
	Sets the height of the slide in points.
Values:	String value of the desired slide height.

Conversion Settings - PowerPoint Page Setup

Name: **Microsoft.PowerPoint.PageSetup.SlideSize**

Sets the slide size for the specified presentation

Values: SlideSizeOnScreen - On Screen
 SlideSizeLetterPaper - Letter Paper
 SlideSizeA4Paper - A4 Paper
 SlideSize35MM - 35MM
 SlideSizeOverhead - Overhead
 SlideSizeBanner - Banner
 SlideSizeLedgerPaper - Ledger Paper
 SlideSizeA3Paper - A3 Paper
 SlideSizeB4ISOPaper - B4 ISO Paper
 SlideSizeB5ISOPaper - B5 ISO Paper
 SlideSizeB4JISPaper - B4 JIS Paper
 SlideSizeB5JISPaper - B5 JIS Paper
 SlideSizeHagakiCard - Hagaki Card

Name: **Microsoft.PowerPoint.PageSetup.SlideWidth**

Sets the width of the slide in points.

Values: String value of the desired slide width.

Conversion Settings - PowerPoint Print Options

Name: **Microsoft.PowerPoint.PrintOptions.FitToPage**

If set to "True" then the slides will be scaled to fill the page they're printed on.

Values: String value "True" or "False".

Name: **Microsoft.PowerPoint.PrintOptions.FrameSlides**

If set to "True" then a thin frame is placed around the border of the printed slides.

Values: String value "True" or "False".

Name: **Microsoft.PowerPoint.PrintOptions.HandoutOrder**

Sets the page layout order for printed handouts that show multiple slides on one page.

Values: PrintHandoutVerticalFirst
 PrintHandoutHorizontalFirst

Name: **Microsoft.PowerPoint.PrintOptions.HighQuality**

If set to "True" then the slides will be printed in high quality.

Values: String value "True" or "False".

Conversion Settings - PowerPoint Print Options

Name: **Microsoft.PowerPoint.PrintOptions.OutputType**

Sets which component (slides, handouts, notes pages, or an outline) of the presentation is to be printed, and in the case of handouts, how many slides per page.

Values: PrintOutputSlides - print slides only.
 PrintOutputNotesPages - prints slides with notes.
 PrintOutputOutline - outline only.
 PrintOutputBuildSlides - build slides only (Office 2003 and 2007 only).
 PrintOutputOneSlideHandouts - handouts with a single slide per page.
 PrintOutputTwoSlideHandouts - handouts with two slides per page.
 PrintOutputThreeSlideHandouts - handouts with three slides per page.
 PrintOutputFourSlideHandouts - handouts with four slides per page.
 PrintOutputSixSlideHandouts - handouts with six slides per page.
 PrintOutputNineSlideHandouts - handouts with nine slides per page.

Name: **Microsoft.PowerPoint.PrintOptions.PrintColorType**

Prints the presentation in one of black and white, in pure black and white (also referred to as high contrast), or in color.

Values: PrintColor
 PrintBlackAndWhite
 PrintPureBlackAndWhite

Name: **Microsoft.PowerPoint.PrintOptions.PrintComments**

If set to "True" then any comments will be printed along with the slides in the presentation.

Values: String value "True" or "False".

Name: **Microsoft.PowerPoint.PrintOptions.PrintFontsAsGraphics**

If set to "True" then any text created with TrueType fonts will be printed as graphics.

Values: String value "True" or "False".

Name: **Microsoft.PowerPoint.PrintOptions.PrintHiddenSlides**

If set to "True" then any hidden slides in the presentation will also be printed.

Values: String value "True" or "False".

Name: **Microsoft.PowerPoint.PrintOptions.SlideShowName**

Sets the name of the custom slide show to print.

Values: A string value containing the name of the custom slide show in the presentation.

Conversion Settings - Document Protection	
Name:	Microsoft.PowerPoint.OpenPassword The password is used to open a password-protected PowerPoint presentation. This password is passed as clear text and is visible to anyone.
Values:	A string value containing the password.

Adobe Reader Options

These options control the behavior of the Adobe Reader converter used by Document Conversion Service. Table values in **bold** text are the default value for that setting.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="TIFF 300dpi Adobe with Markup"
  Description="Prints Adobe files with any stamps and markup visible.">
  <Settings>

    <add Name="Adobe.PDF.PrintOptions.CommentsAndForms"
      Value="DocumentsAndMarkups"/>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name="Save;Output File Format" Value="TIFF Multipaged"/>
    ...

  </Settings>
</Profile>
```

Conversion Settings - Adobe Reader Print Options

Name: **Adobe.PDF.PrintOptions.CommentsAndForms**

Choose what is visible on the page when the PDF file is printed. Markup consists of any comments and annotations, including stamps, that have been placed on the PDF.

Values: **DocumentsAndMarkups** - prints the document with any markup and stamps visible.
 DocumentsAndStamps - prints the document with only stamp annotations visible. Markup is not shown
 Documents - prints only the document. Markup and stamps are not printed.

Name: **Adobe.PDF.PrintOptions.ChoosePaperSourceByPDFPageSize**

When "True", Adobe will use the page size of each page in the PDF to determine the paper size of the output page (paper source); in this case the page size of the output images will match the original PDF document. If you are controlling the paper size using the [Devmode settings;Paper Size](#) setting, this option should be set to *false*. This will tell Adobe to scale the pages to the new paper size. This option is enabled (set to "True") by default.

Values: String value "True" or "False".

Name: **Adobe.PDF.PrintOptions.PageAutoRotate**

When "True", the PDF page will be rotated to fit the output page orientation when needed. Use when *Adobe.PDF.PrintOptions.ChoosePaperSourceByPDFPageSize* is set to "False". This option is disabled (set to "False") by default.

Values: String value "True" or **"False"**.

Conversion Settings - Adobe Reader Print Options

Name:	Adobe.PDF.PrintOptions.PageScaling
	Choose how the PDF page will be scaled to the output page. Use when <i>Adobe.PDF.PrintOptions.ChoosePaperSourceByPDFPageSize</i> is set to "False". This option is set to "ShrinkToFit" by default.
	Note: This option applies only when using Adobe Reader with the Adobe Reader converter; if using Adobe Acrobat, this option is not recognized.
Values:	ActualSize - prints the PDF page at its original page size. If the output page is smaller than the original PDF page size, the page may be cropped. ShrinkToFit - PDF pages that are larger than the output page size will be scaled to fit on the page; smaller pages are not scaled and are centered on the larger page. This is the default value.
Name:	Adobe.PDF.PrintOptions.PrintAsImage
	Choose how the PDF page will be printed. This option is enabled (set to "True") by default as it produces the best quality output.
Values:	String value "True" or "False".
Name:	Adobe.PDF.PrintOptions.PrintCommentPopups
	Set to true to also print comment popups when printing with <i>Adobe.PDF.PrintOptions.CommentsAndForms</i> set to DocumentsAndMarkups. The comments must be open to be printed, otherwise only the comment icon is printed. Valid for Adobe Reader version 10 and higher.
Values:	String value "True" or "False".
Name:	Adobe.PDF.PrintOptions.AllowDuplexPrintJobs
	Allows PDF files set with duplex printing options to successfully convert.
	An empty blank page created by the Adobe printing engine will be added to the end of any documents with an odd number of pages. Setting to "False" will cause the file to fail to convert.
Values:	String value "True" or "False".
Name:	Adobe.PDF.PrintOptions.IgnoreDuplexPrintingOptions
	Ignores any duplex (double-sided, FlipOnLongEdge, FlipOnShortEdge) printing options set in the PDF file. The file is converted single-sided.
	Overrides the Adobe.PDF.PrintOptions.AllowDuplexPrintJobs setting, if set. Does not apply to password-protected PDF files.
Values:	String value "True" or "False".

Conversion Settings - Adobe Reader JavaScript Options**Name:** **Adobe.PDF.Javascript.Enable**

Enable or disable any JavaScript in the PDF document. This option is disabled (set to "False") by default as JavaScript in PDF files can be a security risk. If your PDF files contain JavaScript that you need to have run to display the file properly, you can enable JavaScript processing by setting this options to "True".

Values: String value "True" or "False".**Conversion Settings - Adobe Reader General Options****Name:** **Adobe.PDF.IgnoreSecurity**

Available starting in DCS 3.0.016.

This setting ignores, if possible, any security and passwords set on the PDF file, allowing the PDF file to be converted. PDF files with both user and owner passwords will still fail to convert. This option is enabled (set to "True") by default.

Values: String value "**True**" or "False".**Name:** **Adobe.PDF.CreateTempCopyOnRetry**

Available starting in DCS 3.0.016.

When *True*, this setting will attempt to copy this PDF to a new temporary PDF for processing. For badly formed PDF files this can sometimes repair issues that prevent the file from opening and/or converting. This option is enabled (set to "True") by default.

Values: String value "**True**" or "False".

These options control the behavior of the Internet Explorer converter used by Document Conversion Service. Table values in **bold** text are the default value for that setting.

There are also application level Internet Explorer settings to control image scaling and browser emulation; see [Application Level Configuration Settings](#) to change these options.



```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
    DisplayName="Internet Explorer TIFF 300dpi"
    Description="Created TIFF from IE with footer only.">
  <Settings>

    <!-- Add footer with URL in center -->
    <add Name="Microsoft.InternetExplorer.PageSetup.Footer"
        Value="&#x00000000&#x00000000&#x00000000"/>

    <!-- Add 0.50 inch margins -->
    <add Name="Microsoft.InternetExplorer.PageSetup.MarginBottom"
        Value="0.50"/>
    <add Name="Microsoft.InternetExplorer.PageSetup.MarginLeft"
        Value="0.50"/>
    <add Name="Microsoft.InternetExplorer.PageSetup.MarginRight"
        Value="0.50"/>
    <add Name="Microsoft.InternetExplorer.PageSetup.MarginTop"
        Value="0.50"/>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name="Save;Output File Format" Value="TIFF Multipaged"/>
    ...

  </Settings>
</Profile>
```

Conversion Settings - Page Setup

Name: **Microsoft.InternetExplorer.PageSetup.Font**

The font to use if printing headers and footers. The font is specified as follows, with text in bold specifying the font name, its point size and the color. The last two options, *font-style: italic*; and *font-weight: bold* are optional and are only to be included if bold, italic, or bold and italic text is wanted.

Values: String value containing the font definition.

font-family: <name>; font-size: <size>pt; color: rgb(0,0,0); font-style: italic; font-weight: bold;

Name: **Microsoft.InternetExplorer.PageSetup.MarginBottom**

The bottom margin in inches. The default is 0.75.

Values: String value of the desired margin height.

Name: **Microsoft.InternetExplorer.PageSetup.MarginLeft**

The left-hand side margin in inches. The default is 0.75.

Values: String value of the desired margin width.

Name: **Microsoft.InternetExplorer.PageSetup.MarginRight**

The right-hand side margin in inches. The default is 0.75.

Values: String value of the desired margin width.

Name: **Microsoft.InternetExplorer.PageSetup.MarginTop**

The top margin in inches. The default is 0.75.

Values: String value of the desired margin height.

Name: **Microsoft.InternetExplorer.PageSetup.PrintBackground**

Determines if background colors and images are printed. By default, they are always printed.

Values: String value "True" or "False".

Conversion Settings - Page Setup

Name: **Microsoft.InternetExplorer.PageSetup.ShrinkToFit**

Determines if the page is scaled to fit on the the printed page. By default it is always printed with Shrink-to-Fit enabled.

By default, the minimum scale factor is 30, meaning the page will shrink to at most 30% of its original size to try and fit the contents on the page. If you need the page to be larger, this scaling factor can be customized in the *Internet Explorer* section in the *ApplicationFactory* section of the Document Conversion Service application configuration file. See also [Application Level Configuration Settings](#).

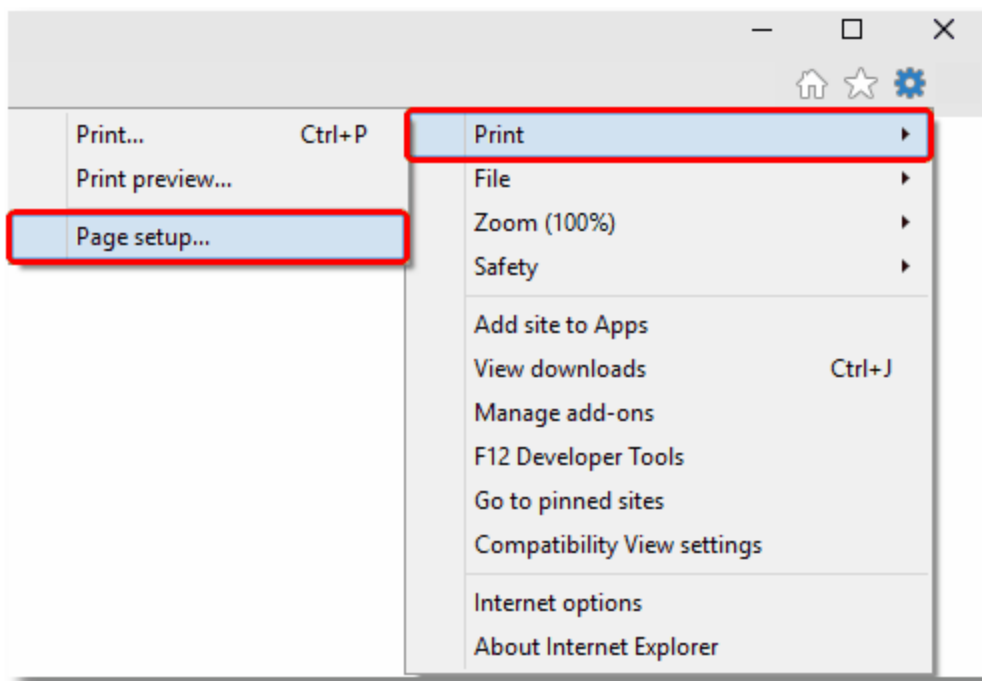
```
<AppFactory Name="Internet Explorer"
  Type="PEERNET.PNDocConv.Applications.PNInternetExplorerApplicationFactory"
  Assembly="PNInternetExplorerApplicationFactory">
  <Settings>
    ...
    <add Name="ConverterPlugIn.PNIExplorer.ShrinkToFitScaleMin" Value="30" />
  </Settings>
</AppFactory>
```

Values: String value "True" or "False".

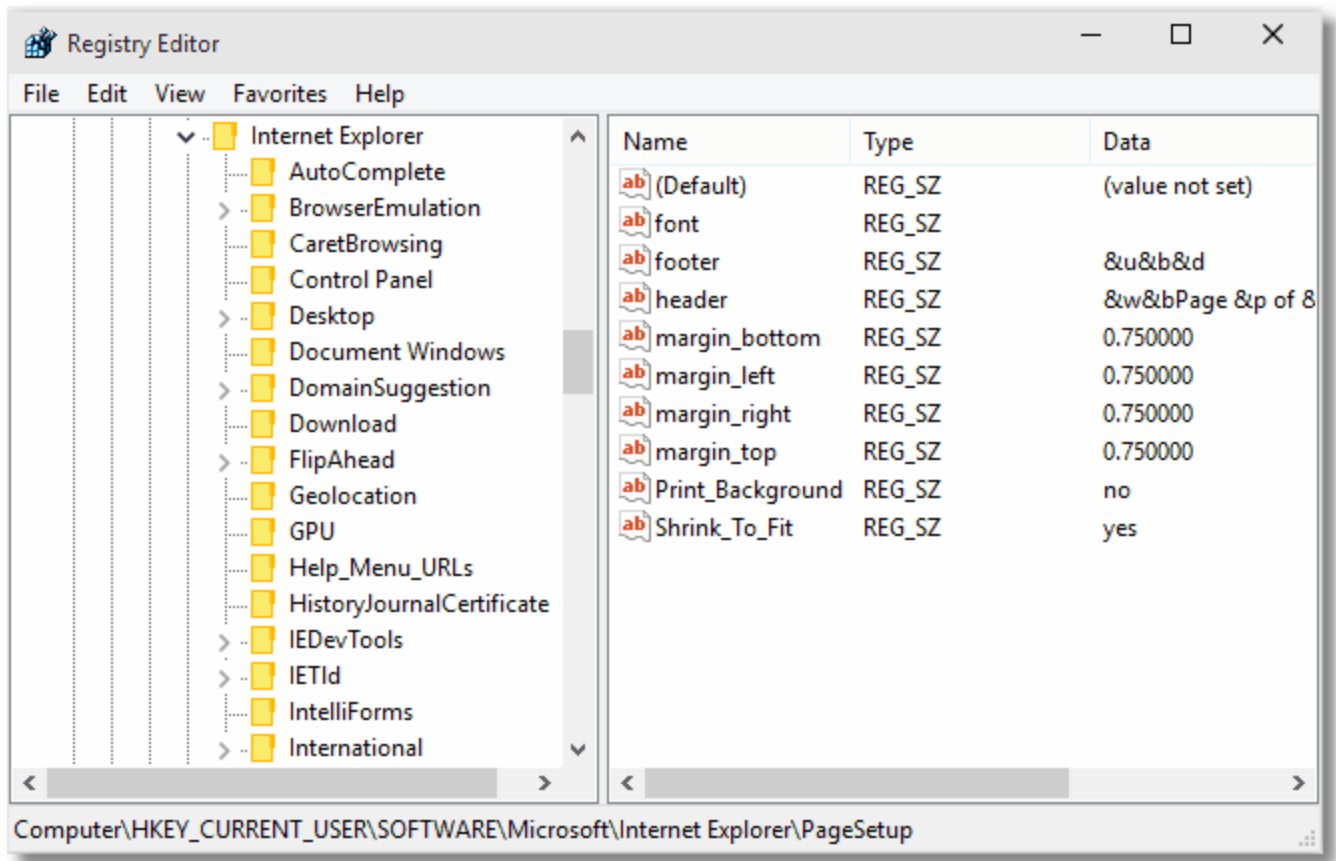
Adding Headers, Footers and Fonts to HTML Conversion

The simplest method to add header and footer information and font information is to use the *Page Setup* dialog in Internet Explorer to configure the margins, headers, footers and other page setup options and then copy these settings from the registry keys Internet Explorer uses to store this information.

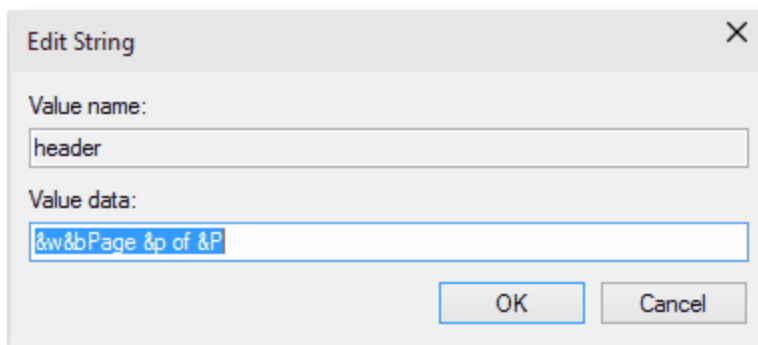
1. Open Internet Explorer to any web page or html file.
2. In the upper right corner, click the Tools icon (it looks like a blue gear), then select Print - Page Setup.
 - a. Alternatively you can press the F10 key to show the application menu and then select File - Page Setup.



3. In the Page Setup dialog, define your margins, any header and footer information, and optionally choose the font you want to use. Click OK, then exit Internet Explorer.
4. Open the registry using *RegEdit* (type regedit.exe into the Start menu search field or from the Start menu go to Programs - Accessories - Run and type regedit.exe).
5. In the registry editor, go to the HKEY_CURRENT_USER folder, then Software - Microsoft - Internet Explorer - PageSetup.



6. In the right-hand pane, double click any of the values to open the *Edit String* dialog box. From here you can copy and paste the header and footer formatted strings. When using these strings in the conversion profiles, any & characters need to be replaced with & for the string to be parsed correctly.



Application Level Configuration Settings

Document Conversion Service uses Internet Explorer to convert HTM, HTML and MHT files. When dealing with MHT and HTML files with large images, and older style HTML files formatted for earlier browser versions the options for image scaling and browser emulation may need to be configured to produce the desired output file.

These options are set in the Internet Explorer section of the application configuration file. Changing these options will require a restart of Document Conversion Service for the new settings to take effect.

Setting the Minimum Scale For Internet Explorer

HTML files and MHT files such as email messages from Outlook can sometimes have very wide images. By default, these files are always printed with Shrink-to-Fit enabled and a minimum scale factor of 30. This means that the page will shrink to at most 30% of its original size to fit the image contents on the page.

If you need the images to be scaled larger, the setting *ConverterPlugIn.PNIEplorer.ShrinkToFitScaleMin* can be adjusted from between 30 to 100 to get the size of image you want.

This option is set at the application level and cannot be changed per file. Changes to this setting require a restart of Document Conversion Service to take effect.

Setting the Browser Emulation for Internet Explorer

In certain cases, older HTML files created for previous versions of Internet Explorer will not convert correctly when printed using the latest version of Internet Explorer. This is because Internet Explorer runs with *Edge compatibility* by default and it is this new compatibility and rendering that has a problem with the older style HTML.

If you have these type of files, the setting *ConverterPlugIn.PNIEplorer.BrowserEmulation* can be used to force Internet Explorer to emulate older versions of the browser so that the files are rendered properly based on the older browsers rendering engine.

This option is set at the application level and cannot be changed per file. Changes to this setting require a restart of Document Conversion Service to take effect.



Configuration Section for Internet Explorer

```
<AppFactories>
  <Factories>

    <AppFactory Name="Internet Explorer"
      Type="PEERNET.PNDocConv.Applications.PNInternetExplorerApplicationFactory"
      Assembly="PNInternetExplorerApplicationFactory">
      <Settings>
        <add Name="Enabled" Value="auto"/>
        <add Name="MaxInstances" Value="auto"/>
        <add Name="RecycleThreshold" Value="0"/>
        <add Name="DocumentOpenTimeout" Value="360000"/>

        <!-- Value range 30 - 100 -->
        <add Name="ConverterPlugIn.PNIEplorer.ShrinkToFitScaleMin" Value="30"/>

        <!-- Values: Empty string, IE7, IE8, IE8FORCE, IE9, IE9FORCE, IE10, IE10FORCE, IE11, IE11FORCE -->
        <add Name="ConverterPlugIn.PNIEplorer.BrowserEmulation" Value="" />
      </Settings>
    </AppFactory>
    ...
  </Factories>
</AppFactories>
```

Ghostscript Converter Options

These options control the behavior of the Ghostscript converter used by Document Conversion Service. Table values in **bold** text are the default value for that setting.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="Ghostscript Custom Font TIFF 300dpi"
  Description ="Created TIFF from Ghostscript using custom fonts.">
  <Settings>

    <!-- Antialias options -->
    <add Name="ConverterPlugIn.PNGhostscriptConverter.TextAntiAlias" Value="4"/>
    <add Name="ConverterPlugIn.PNGhostscriptConverter.GraphicsAntiAlias" Value="4"/>

    <!-- Search these folders for matching fonts -->
    <add Name="ConverterPlugIn.PNGhostscriptConverter.FontPath"
      Value="C:\psfonts;c:\Windows\Fonts;C:\MyFonts"/>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name="Save;Output File Format" Value="TIFF Multipaged"/>
    ...

  </Settings>
</Profile>
```

Name:	ConverterPlugIn.PNGhostscriptConverter.TextAntiAlias
	The size of the subsample box used when antialiasing text in the file. Antialiasing is used to improve the quality of the text on the page when converted to an image. A subsample box of 4 will produce the best result. The lower subsample values will increase the speed of conversion but can affect the image quality.
Values:	The size of the subsample box can be 4, 2 or 1. The default is 4 .
Name:	ConverterPlugIn.PNGhostscriptConverter.GraphicsAntiAlias
	The size of the subsample box used when antialiasing graphics in the file. Antialiasing is used to improve the quality of any graphics on the page when converted to an image of a different resolution. A subsample box of 4 will produce the best result. The lower subsample values will increase the speed of conversion but can affect the image quality.
Values:	The size of the subsample box can be 4, 2 or 1. The default is 4 .
Name:	ConverterPlugIn.PNGhostscriptConverter.FontPath
	By default, the special Windows <i>Fonts</i> folder and the folder c:\psfonts are used by Ghostscript to find the fonts used in the Postscript or PDF documents. You can override this setting by providing your own semicolon-separated list of folders in which to search.
Values:	String value containing a semi-colon separated list of folders.

Image Converter Options

These options control the behavior of the image converter used by Document Conversion Service. Table values in **bold text** are the default value for that setting.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="Image Converter 300dpi"
  Description="Create images using the PEERNET image converter.">
  <Settings>

    <!-- Image Conversion options, use LEAD first -->
    <add Name="ConverterPlugIn.PNImageConverter.ImageToolkitOrder" Value="LEAD;WIC"/>
    <add Name="ConverterPlugIn.PNImageConverter.LEADScalingMode" Value="BICUBIC"/>
    <add Name="ConverterPlugIn.PNImageConverter.WICScalingMode" Value="BICUBIC"/>

    <!-- Background color for transparent images, white is default -->
    <add Name="ConverterPlugIn.PNImageConverter.AlphaBackgroundColorRGB" Value="255,255,255"/>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name="Save;Output File Format" Value="TIFF Multipaged"/>
    ...

  </Settings>
</Profile>
```

Conversion Settings - Toolkits and Scaling Modes

Name: ConverterPlugIn.PNImageConverter.ImageToolkitOrder

This string lists, in the order in which they will be used, the image tool kits that PEERNET Image Converter will use to try and convert an image. The default value, "LEAD;WIC", will use LEAD first and then try WIC (**W**indows **I**maging **C**omponent) if the image could not be converted. The two tool kits support opening and reading different file formats; see [Supported Image File Formats](#) below for a complete list. You do not need to install anything extra to use these either of these tool kits. The LEAD tool kit is bundled with Document Conversion Service and the Windows Image Component is part of the Windows operating system.

Values: **LEAD;WIC** - use LEAD first, then try WIC if the image could not be converted.
 WIC;LEAD - use WIC first, then try LEAD if the image could not be converted.
 LEAD - only use LEAD.
 WIC - only use WIC.

Name: ConverterPlugIn.PNImageConverter.LEADScalingMode

This is the sampling or filtering mode to use when scaling an image. An image needs to be scaled when the resolution of the source image and destination image are not the same.

Values: **NORMAL** - Nearest neighbor, this is the fastest mode and often can produce the smallest image.
LINEAR - A linear interpolation algorithm, slower than NORMAL but better image quality.
BICUBIC - Bicubic interpolation resizing, slower than LINEAR, but better image quality.

Conversion Settings - Toolkits and Scaling Modes

Name: **ConverterPlugIn.PNImageConverter.WICScalingMode**

This is the sampling or filtering mode to use when scaling an image. An image needs to be scaled when the resolution of the source image and destination image are not the same.

Values: NORMAL - Uses nearest neighbor scaling. This is nearest neighbor scaling, which is the fastest mode and often can produce the smallest image. The tradeoff is a lower image quality.
 LINEAR - A bilinear interpolation algorithm where the weighted average of a 2x2 grid is used to compute the pixel values of the new image. Better quality than NORMAL.
BICUBIC - The new pixel values are computed using a weighted average of a 4x4 grid.
 FANT - This scaling mode produces the best quality images but is slower and more CPU intensive than the others.

Name: **ConverterPlugIn.PNImageConverter.KeepSourceImageResolution**

Optionally keep the output image's resolution the same as source image. Note that fax mode and other image option actions ([Image Options](#)) will still override the end result. Overrides the *Devmode settings;Resolution* settings from [Devmode settings](#).

Values: True - Create the new image with the same resolution as the original image.
 False - Creates the new image with the resolution specified in the *Devmode settings;Resolution* setting.

Name: **ConverterPlugIn.PNImageConverter.ResampleImageToMaxWidthOrHeightInPixels**

Dynamically sample the output image to a specific maximum width or height, which ever criteria is met first. The desired dimension is specified in *pixels*. Note that fax mode and other image option actions ([Image Options](#)) will still override the end result.

Values: The desired maximum width or height in pixels.

Name: **ConverterPlugIn.PNImageConverter.AlphaBackgroundColorRGB**

For images that support transparency, or alphablending, optionally set the desired background color when converting the image. The default background color is White.

Values: The desired background color set as RGB triplet separated by commas.

255,255,255 - White
0,0,0 - Black

Supported Image File Formats

The table below lists the image formats supported by each tool kit.

Cserve Portable Network Graphics images (*.png)	•	•
Graphics Interchange Format image files (*.gif)	•	•
Icon Format (*.ico)		•
JPEG images (*.jpg)	•	•

TIFF images (*.tif)	•	•
Windows Bitmap images (*.bmp)	•	•
Windows Media Photo (*.wdp, *.hdp, *.jxr)		•
ZSoft PCX images (*.pcx)	•	
ZSoft DCX images (*.dcx)	•	

OutsideIn AX Options

These options control the behavior of the OutsideIn AX converter used by Document Conversion Service. Table values in **bold** text are the default value for that setting.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="TIFF 300dpi OutsideIn AX"
  Description ="OutsideIn AX with borders and margins">
  <Settings>
    <add Name ="Oracle.OutsideInAX.BMPPrintBorder" Value="0"/>
    <add Name ="Oracle.OutsideInAX.IntlFlags" Value="1"/>
    <add Name ="Oracle.OutsideInAX.PrintMarginTop" Value="0.50"/>
    <add Name ="Oracle.OutsideInAX.PrintMarginBottom" Value="0.50"/>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name="Save;Output File Format" Value="TIFF Multipaged"/>
    ...
  </Settings>
</Profile>
```

Conversion Settings - OutsideIn AX Printing	
Name:	Oracle.OutsideInAX.BMPPrintBorder
	Print a one pixel wide border around the image.
Values:	0 - do not print the border 1 - print the border
Name:	Oracle.OutsideInAX.VECPrintBorder
	Print a one pixel wide border around the image.
Values:	0 - do not print the border 1 - print the border
Name:	Oracle.OutsideInAX.IntlFlags
	Specifies what unit of measurement is used for the print margins below. Units are either inches or metric units.
Values:	0 - Metric 1 - Imperial (Inches)
Name:	Oracle.OutsideInAX.PrintMarginTop
	The top print margin height.
Values:	A string value representing the printer margin as a floating point number, such as 0.50 for half an inch.

Conversion Settings - OutsideIn AX Printing**Name:** **Oracle.OutsideInAX.PrintMarginBottom**

The bottom print margin height.

Values: A string value representing the printer margin as a floating point number, such as 0.50 for half an inch.**Name:** **Oracle.OutsideInAX.PrintMarginLeft**

The left print margin width.

Values: A string value representing the printer margin as a floating point number, such as 0.50 for half an inch.**Name:** **Oracle.OutsideInAX.PrintMarginRight**

The right print margin width.

Values: A string value representing the printer margin as a floating point number, such as 0.50 for half an inch.

Save

These options control the orientation, resolution, color mode and paper size of the output file. You can also choose to split multipage files based on the number of pages per file or a file size threshold. Table values in **bold** text are the default value for that setting.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="TIFF 300dpi Serialized with Overwrite"
  Description ="Create TIFF serialized, overwrite existing files.">
  <Settings>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name ="Save;Output File Format" Value="TIFF Serialized"/>
    <add Name ="Save;Prompt" Value="0" />
    <add Name ="Save;Overwrite" Value="1"/>
    <add Name ="Save;Color reduction" Value="BW"/>
    ...

  </Settings>
</Profile>
```

Conversion Settings - Save	
Name:	Save;Use JobID
	Use the driver JobID when creating the file name. The driver stores an internal number that is automatically incremented for each print job.
Values:	0 - Do not include JobID in file name. 1 - Include JobID in file name.
Name:	Save;Append
	Append the new images to an existing file name or sequence.
Values:	0 - Do not append, output is a new file. 1 - Output is appended to existing file or sequence.
Name:	Save;Output directory
Values:	The output directory path in which to save the image.
Name:	Save;Output filename
Values:	Base file name excluding path and extension to use to name the file. Default is the document name submitted to print job.

Conversion Settings - Save

Name: Save;Output File Format

The type of file to create.

Values: **JPEG** - JPEG (*.jpg)
 TIFF Multipaged - TIFF Multipaged (*.tif)
 TIFF Serialized - TIFF Serialized (*.tif)
 Adobe PDF Multipaged - Adobe PDF Multipaged (*.pdf)
 Adobe PDF Serialized - Adobe PDF Serialized (*.pdf)
 CompuServe GIF - CompuServe GIF (*.gif)
 CompuServe PNG - CompuServe PNG (*.png)
 Windows BMP - Windows BMP (*.bmp)
 TARGA - Targa (*.tga)
 Adobe Photoshop 3.0 - Adobe Photoshop 3.0 (*.psd)
 ZSoft PCX - ZSoft PCX (*.pcx)
 ZSoft DCX - ZSoft DCX (*.dcx)

Name: Save;remove file extension

Removes the filename extension from the original filename before creating the new filename. If set to 0, a file *Document.doc* created as TIFF would become *Document.doc.tif*, when set to remove the extension, the resulting filename would be *Document.tif*.

Values: **0** - Leave original filename extension in new filename
 1 - Remove original filename extension before creating new filename.

Name: Save;Color reduction

Use the color reduction options below to reduce the number of colors in the output files.

Values: none - No color reduction
Optimal - Reduce to lowest color count needed per page
 BW - Reduce to black and white using selected dithering method
 grey - Reduce to greyscale
 256Colors - Create all pages as 8-bit color (256 colors)
 16Colors - Create all pages as 4-bit color (16 colors)
 optimalMax256Colors - Reduces to lowest color count needed for each page, any pages over 256 colors are reduced to 256 colors.
 optimalMax16Colors - Reduces to lowest color count needed for each page, any pages over 16 colors are reduced to 16 colors.

Name: Save;Dithering method

Dithering enhances the appearance of color images that have been reduced to black and white.

Values: None - No dithering
 Floyd - Floyd-Steinberg dithering
 Burkes - Burkes dithering
 Bayer - Bayer dithering
Halftone - Halftone dithering

Conversion Settings - Save

Name: **Save;SplitFileEveryNPagesEnabled**

Enables file splitting based on the page count set by **SplitFileEveryNPages**. When file splitting is enabled, the [serialized naming profile](#) is always used to name each file in the sequence. Can be combined with *SplitFileWhenFileSizeExceedsThresholdEnabled* to split by page count and file size.

File splitting only applies to the following multipaged file formats:

- TIFF Multipaged - TIFF Multipaged (*.tif)
- Adobe PDF Multipaged - Adobe PDF Multipaged (*.pdf)
- ZSoft DCX - ZSoft DCX (*.dcx)

Values: **0** - Do not split the file, create a single multipaged file.
1 - Split the file when the page count reaches limit set by SplitFileEveryNPages.

Name: **Save;SplitFileEveryNPages**

The page count at which to start creating a new file.

Values: 0-4294967295, default is **1000**.

Name: **Save;SplitFileWhenFileSizeExceedsThresholdEnabled**

Enables file splitting based on a file size threshold set by **SplitFileSizeThresholdInBytes**. The file is split when the file size gets larger than the threshold. When file splitting is enabled, the [serialized naming profile](#) is always used to name each file in the sequence. Can be combined with *SplitFileEveryNPagesEnabled* to split by file size and page count.

File splitting only applies to the following multipaged file formats:

- TIFF Multipaged - TIFF Multipaged (*.tif)
- Adobe PDF Multipaged - Adobe PDF Multipaged (*.pdf)
- ZSoft DCX - ZSoft DCX (*.dcx)

Values: **0** - Do not split the file, create a single multipaged file.
1 - Split the file when the file size exceeds the limit set by SplitFileSizeThresholdInBytes.

Name: **Save;SplitFileSizeThresholdInBytes**

The file size, in bytes, at which to start creating a new file.

Values: 0-18446744073709551615, default is **1073741824**, or 1GB.

Devmode settings

These options control the orientation, resolution, color mode and paper size of the output file. Table values in **bold** text are the default value for that setting.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="TIFF 300dpi Color "
  Description ="Create color TIFF">
  <Settings>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name="Devmode settings;Color" Value="1"/>
    <add Name="Save;Output File Format" Value="TIFF Multipaged"/>
    ...

  </Settings>
</Profile>
```

Conversion Settings - Devmode

Name: **Devmode settings;Orientation**

Orientation of the page when the file is converted.

Values: **Portrait**
Landscape

Name: **Devmode settings;Resolution**

Number of dots per inch.

Values: 1200, 720, 600, 400, 360, **300**, 254, 240, 200, 150, 120, 100, 75, 60, 50

Name: **Devmode settings;Color**

Print files in color or black and white

Values: **1**
Color mode
0
Black and white, or monochrome mode.

Name: **Devmode settings;Paper Size**

Standard paper sizes available. Other custom paper sizes you may have added are also available by name.

Values: **Letter**
Letter Small
Tabloid
Legal
Statement
Executive
A3
A4

Conversion Settings - Devmode**Name: Devmode settings;Paper Size**

Standard paper sizes available. Other custom paper sizes you may have added are also available by name.

A4 Small
A5
B4
B5
Folio
Quarto
10x14
11x17
Note
Envelope #9
Envelope #10
Envelope #11
Envelope #12
Envelope #14
C Size Sheet
D Size Sheet
E Size Sheet
F Size Sheet
Envelope DL
Envelope C5
Envelope C3
Envelope C4
Envelope C6
Envelope C65
Envelope B4
Envelope B5
Envelope B6
Envelope Italy
Envelope Monarch
Envelope Personal
US Std Fanfold
German Std Fanfold
German Legal Fanfold
ISO B4
Japanese Postcard
9x11
10x11
15x11
Envelope Invite
Letter Extra
Legal Extra
Tabloid Extra
A4 Extra
Letter Transverse
A4 Transverse
Letter Extra Transverse
A Plus
B Plus
Letter Plus
A4 Plus
A5 Transverse
B5 Transverse
A3 Extra
A5 Extra
B5 Extra

Conversion Settings - Devmode**Name: Devmode settings;Paper Size**

Standard paper sizes available. Other custom paper sizes you may have added are also available by name.

A3 Transverse
A3 Extra Transverse
A1 594 x 841 mm
A0 841 x 1189 mm
B3 (ISO) 353 x 500 mm
B2 (ISO) 500 x 707 mm
B1 (ISO) 707 x 1000 mm
B3 (JIS) 364 x 515 mm
B2 (JIS) 515 x 728 mm
B1 (JIS) 728 x 1030 mm
B0 (JIS) 1030 x 1456 mm

Advanced File Naming

There are four different naming profiles that control how the output file is named. Which naming profile is used depends on if you are creating serialized or multipaged output, and if you have the Save;UseJobID setting set to true. It is the combination of these settings that determines which profile is used to build the output filename.

The only exception to this is when file splitting by page count (Save;SplitFileEveryNPagesEnabled) or file size (Save;SplitFileWhenFileSizeExceedsThresholdEnabled) is enabled. When file splitting is enabled, the serialized naming profile is always used to name each file in the sequence. The file splitting options are only used when creating multipaged file types.

Serialized or Multi-page	Include JobID	Naming Profile
Serialized	No	Serialized
	Yes	Serialized w/ JobID
Multi-paged	No	Multi-page
	Yes	Multi-page w/ JobID

In most scenarios you will never need to change these values. Care must be taken when you do. The table below lists the settings to use to customize the output file naming. Table values in **bold** text are the default value for that setting.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="Color TIFF 300dpi Named by Page Number"
  Description ="Create color TIFF named by page number">
  <Settings>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name="Devmode settings;Color" Value="1"/>
    <add Name="Save;Output File Format" Value="TIFF Serialized"/>
    <add Name="Advanced File Naming;Format string S" Value="%s"/>
    <add Name="Advanced File Naming;Variables S"
      Value="$(PrintedPageNumber)"/>
    ...
  </Settings>
</Profile>
```

Conversion Settings - Advanced File Naming

Name:

Advanced File Naming;Format string S

Format string for the serialized naming profile. Also used to name the sequence of files when file splitting is enabled.

Values:

A string containing the format string used to create the output file name. The format string can contain placeholders %s and %d that correspond to the variables passed in *Advanced File Naming;Variables S* below.

Conversion Settings - Advanced File Naming

Name: **Advanced File Naming;Use default extension S**

Use the default file extension for the output type when naming the output file.

Values: 0 - Do not use default file extension
1 - Use default file extension

Name: **Advanced File Naming;Variables S**

Comma-delimited list of variables that correspond to the placeholders in the format string supplied in *Advanced File Naming;Format string S* above.

Values: See list of [variables](#) below.

Name: **Advanced File Naming;Format string SJ**

Format string for serialized with JobID naming profile. In this profile a JobID, a number that is automatically incremented, is used as part of the filename.

Values: A string containing the format string used to create the output file name. The format string can contain placeholders %s and %d that correspond to the variables passed in *Advanced File Naming;Variables SJ* below.

Name: **Advanced File Naming;Use default extension SJ**

Use the default file extension for the output type when naming the output file.

Values: 0 - Do not use default file extension
1 - Use default file extension

Name: **Advanced File Naming;Variables SJ**

Comma-delimited list of variables that correspond to the placeholders in the format string supplied in *Advanced File Naming;Format string SJ* above.

Values: See list of [variables](#) below.

Name: **Advanced File Naming;Format string M**

Format string for the multipaged naming profile.

Values: A string containing the format string used to create the output file name. The format string can contain placeholders %s and %d that correspond to the variables passed in *Advanced File Naming;Variables M* below.

Name: **Advanced File Naming;Use default extension M**

Use the default file extension for the output type when naming the output file.

Values: 0 - Do not use default file extension
1 - Use default file extension

Conversion Settings - Advanced File Naming**Name: Advanced File Naming;Variables M**

Comma-delimited list of variables that correspond to the placeholders in the format string supplied in *Advanced File Naming;Format string M* above.

Values: See list of [variables](#) below.

Name: Advanced File Naming;Format string MJ

Format string for the multipaged with JobID naming profile. In this profile a JobID, a number that is automatically incremented, is used as part of the filename.

Values: A string containing the format string used to create the output file name. The format string can contain placeholders %s and %d that correspond to the variables passed in *Advanced File Naming;Variables MJ* below.

Name: Advanced File Naming;Use default extension MJ

Use the default file extension for the output type when naming the output file.

Values: 0 - Do not use default file extension
1 - Use default file extension

Name: Advanced File Naming;Variables MJ

Comma-delimited list of variables that correspond to the placeholders in the format string supplied in *Advanced File Naming;Format string MJ* above.

Values: See list of [variables](#) below.

Variables for Custom Naming

Variable	Type and Format String Place Holder	Description
\$(Day)	Numeric, %d	The day in numeric format that the print job was submitted to the printer, from 1-31.
\$(DocumentPageNumber)	Numeric, %d	The page number of the document being printed.
\$(FileExtension)	String, %s	The file extension for the type of file being created.
\$(FileNumber)	Numeric, %d	The file number of the sequence of files. For multipaged output, this is always 1. For serialized output this is the number of the file in the sequence.
\$(Hour)	Numeric, %d	The hour in numeric format that the print job was submitted to the printer, 1-12 or 0-23 depending on your system preferences.

Variable	Type and Format String Place Holder	Description
\$(JobID)	Numeric, %d	The unique JobID used by the printer. This is set to zero when the driver is first installed and is automatically incremented by the driver at the start of every print job. The JobID is often used to ensure that all files created have unique names.
\$(JobStatus)	Numeric, %d	The status of the print job, 1 for success, 0 for failure.
\$(MachineName)	String, %s	The name of the computer the print job is running on.
\$(Minute)	Numeric, %d	The minute in numeric format that the print job was submitted to the printer, from 0-59.
\$(Month)	Numeric, %d	The month in numeric format that the print job was submitted to the printer, from 1-12.
\$(OutputFileName)	String, %s	The contents of the \$(OutputFileName) field. If this field is empty the name the printing application used when submitting the print job is used.
\$(PrintedPageNumber)	String, %s	The page number of the page being printed; this is not always the same as \$(DocumentPageNumber).
\$(Second)	Numeric, %d	The second in numeric format that the print job was submitted to the printer, from 0-59.
\$(UserName)	String, %s	The name of the user who submitted the print job.
\$(Year)	Numeric, %d	The year in numeric format that the print job was submitted to the printer.

Default Naming Profile Strings

Profile	Format String	Variables and Resulting File Names for TIFF Creation
Serialized	%s_%3d	\$(OutputFileName) \$(FileName) C:\Test\Invoice_001.tif C:\Test\Invoice_002.tif C:\Test\Invoice_003.tif ...
Serialized w/ JobID	%3d_%s_%3d	\$(JobID) \$(OutputFileName) \$(FileName) C:\Test\010_Invoice_001.tif C:\Test\010_Invoice_002.tif C:\Test\010_Invoice_003.tif ...
Multi-page	%s	\$(OutputFileName) C:\Test\Invoice.tif
Multi-page w/ JobID	%3d_%s	\$(JobID) \$(OutputFileName) C:\Test\011_Invoice.tif

Image Options

These options control the fax mode and creation of the output file. Table values in **bold** text are the default value for that setting.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="Create Fax TIFF Serialized"
  Description ="Create profile F fax 204 x 196">
  <Settings>

    <!-- Output file options -->
    <add Name="Devmode settings;Color" Value="1"/>
    <add Name="Image Options;Fax" Value="1"/>
    <add Name="Image Options;Fax Profile" Value="0"/>
    <add Name="Image Options;Fax Resolution" Value="3"/>
    <add Name="Save;Output File Format" Value="TIFF Serialized"/>
    ...

  </Settings>
</Profile>
```

Conversion Settings - Image Options	
Name:	Image Options;Fax
Values:	0 - Do not create fax format file. 1 - Create an image where its width is limited to fax resolution as determined by Fax Profile and Fax Resolution settings
Name:	Image Options;Fax Profile
Values:	0 - Profile F, standard monochrome 1 - Profile S, simplified monochrome 2 - Profile C, color fax
Name:	Image Options;Fax Resolution
Values:	0 - 200 x 100 resolution (Profile S, F) 1 - 200 x 200 resolution (Profile S, F, C) 2 - 204 x 98 resolution (Profile S, F) 3 - 204 x 196 resolution (Profile S, F) 4 - 300 x 300 resolution (Profile F, C) 5 - 400 x 400 resolution (Profile F, C) 6 - 408 x 391 resolution (Profile F) 7 - 204 x 391 resolution (Profile F) 8 - 300 x 600 resolution (Profile F) 9 - 400 x 800 resolution (Profile F) 10 - 600 x 600 resolution (Profile F, C) 11 - 600 x 1200 resolution (Profile F) 12 - 1200 x 1200 resolution (Profile F, C) 13 - 100 x 100 resolution (Profile F, C)

Conversion Settings - Image Options

Name: Image Options;Fax Use Printer Resolution

Values: 0 - Do not use printer resolution
1 - Use printer resolution

Name: Image Options;Fax Paper Width

Values: 0 - Letter
1 - Legal
2 - A4 (ISO)
3 - B4 (ISO)
4 - A3 (ISO)
5 - Auto

Name: Image Options;Fax Paper Height

Values: 0 - Variable height
1 - Fixed height

Name: Image Options;Fax Page Scaling

Values: 0 - Fit to Page
1 - Actual Size

Name: Image Options;Fax Page Scaling Auto Rotate

Values: 0 - Do not auto-rotate the page
1 - Auto-rotate the page if needed

Name: Image Options;Fax Page Scaling Lock Aspect Ratio

Values: 0 - Do not maintain fax page aspect ratio when scaling
1 - Maintain fax page aspect ratio when scaling

Name: Image Options;Fax Page Scaling Shrink Larger

Values: 0 - Do not shrink fax to fit on page
1 - Shrink fax to fit on page

Name: Image Options;Fax Page Scaling H Align

Values: Left - Align image left
Middle - Align image in the center
Right - Align image right

Conversion Settings - Image Options

Name:	Image Options;Fax Page Scaling V Align
Values:	Top - Align image top Middle - Align image in the center Bottom - Align image bottom
Name:	Image Options;Fax Page Use 256 Greyscale Palette
Values:	0 - Use the smaller 64 grayscale palette 1 - Use 256 grayscale palette
Name:	Image Options;Fill order
Values:	LSB2MSB - Least significant bit to most significant bit MSB2LSB - Most significant bit to least significant bit
Name:	Image Options;EOLs Byte Aligned
Values:	0 - EOLs not byte aligned (no fillbits) 1 - EOLs byte aligned (use fillbits)
Name:	Image Options;Photometric
Values:	MinIsWhite MinIsBlack
Name:	Image Options;Include DateTime
Values:	0 - DateTime field not included in file 1 - DateTime field included in file
Name:	Image Options;Motorola Format
Values:	0 - Use Intel byte order 1 - Use Motorola byte order
Name:	Image Options;Rotate portrait
	Specified in degrees of rotation (counter-clockwise).
Values:	0 90 180 270

Conversion Settings - Image Options**Name:** Image Options;Rotate landscape

Specified in degrees of rotation (counter-clockwise).

Values: 0
90
180
270**Name:** Image Options;Include Software Name and Release**Values:** 0 - Software field not included in file
1 - Software field field included in file

TIFF File Format

Table values in **bold** text are the default value for that setting.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="Create Compressed TIFF Serialized"
  Description ="Create Compressed TIFF Serialized">
  <Settings>

    <!-- Output file options -->
    <add Name="Devmode settings;Color" Value="1"/>
    <add Name="Save;Output File Format" Value="TIFF Serialized"/>
    <add Name="Save;Color reduction" Value="Optimal"/>
    <add Name="TIFF File Format;BW compression" Value="Group3-2D"/>
    <add Name="TIFF File Format;Color compression" Value="LZW"/>
    ...

  </Settings>
</Profile>
```

Conversion Settings - TIFF File Format	
Name:	TIFF File Format; BW compression
Values:	None - No black and white compression Group4 - CCITT Group4 Fax compression Group3-2D - CCITT Group3 2D Fax compression Group3-1D - CCITT Group3 1D Fax compression MH - CCITT Modified Huffman compression LZW - LZW compression Packbits - Packbits (RLE) compression
Name:	TIFF File Format; Color compression
Values:	Uncompressed RGB - No color compression Uncompressed CMYK - No color compression, CMYK color Packbits RGB -Packbits (RLE) compression Packbits CMYK -Packbits (RLE) compression, CMYK color High quality JPEG - High quality JPEG compression Medium quality JPEG - Medium quality JPEG compression Low quality JPEG - Low quality JPEG compression LZW RGB - LZW compression LZW CMYK - LZW compression, CMYK color
Name:	TIFF File Format; Indexed compression
Values:	Uncompressed - No color compression Packbits - Packbits (RLE) compression High quality JPEG - High quality JPEG compression Medium quality JPEG - Medium quality JPEG compression Low quality JPEG - Low quality JPEG compression LZW - LZW compression

Conversion Settings - TIFF File Format**Name:** **TIFF File Format;Greyscale compression****Values:** Uncompressed - No color compression
Packbits - Packbits (RLE) compression
High quality JPEG - High quality JPEG compression
Medium quality JPEG - Medium quality JPEG compression
Low quality JPEG - Low quality JPEG compression
LZW - LZW compression

PDF File Format

These options control the compression methods used during the creation of PDF output files. Table values in **bold** text are the default value for that setting.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="Create PDF/A "
  Description ="Create PDF/A-1b">
  <Settings>

    <!-- Output file options -->
    <add Name ="Devmode settings;Color" Value="1"/>
    <add Name ="Save;Output File Format" Value="PDF Multipaged"/>
    <add Name ="Save;Prompt" Value="0"/>
    <add Name ="Save;Overwrite" Value="1"/>
    <add Name ="PDF File Format;PDF Standard" Value="PDF/A-1b"/>
    <add Name ="PDF File Format;Use compression" Value="1"/>
    ...
  </Settings>
</Profile>
```

Conversion Settings - PDF File Format	
Name:	PDF File Format;Embed Pages as Images
Values:	0 - Creates vector pages, where possible, in the PDF file; does not OCR 1 - Embeds each page of the PDF as an image, creating a raster PDF
Name:	PDF File Format;Include Outline
	This setting applies only when creating vector PDF files, and only if the source file contains outline information. Outline information is shown as bookmarks in a PDF document.
Values:	0 - Does not include outline information in vector PDF files 1 - Includes outline (heading) information, where possible, in vector PDF files
Name:	PDF File Format;Use compression
Values:	0 - Do not compress the file 1 - Enable compression for the file
Name:	PDF File Format;Use ASCII
Values:	0 - No ASCII format compression 1 - Enable ASCII format compression
Name:	PDF File Format;PDF Standard
Values:	None - Create PDF files that are not PDF/A-1b compliant PDF/A-1b - Create PDF/A-1b compliant PDF files when creating raster PDF

Conversion Settings - PDF File Format

Name: PDF File Format;Content encoding

Values: None - No compression
 ZIP - ZIP compression
 RLE - Packbits (run length) compression
LZW - LZW compression

Name: PDF File Format;Color compression

Values: None - No color compression
 ZIP - ZIP compression
 RLE - Packbits (run length) compression
 JPEG High - High quality JPEG compression
 JPEG Medium - Medium quality JPEG compression
 JPEG Low - Low quality JPEG compression
LZW - LZW compression

Name: PDF File Format;Greyscale compression

Values: None - No color compression
 ZIP - ZIP compression
 RLE - Packbits (run length) compression
 JPEG High - High quality JPEG compression
 JPEG Medium - Medium quality JPEG compression
 JPEG Low - Low quality JPEG compression
LZW - LZW compression

Name: PDF File Format;Indexed compression

Values: None - No color compression
 ZIP - ZIP compression
 RLE - Packbits (run length) compression
 JPEG High - High quality JPEG compression
 JPEG Medium - Medium quality JPEG compression
 JPEG Low - Low quality JPEG compression
LZW - LZW compression

Name: PDF File Format;BW compression

Values: None - No black and white compression
Group4 - CCITT Group4 Fax compression
 Group3-2D - CCITT Group3 2D Fax compression
 Group3-1D - CCITT Group3 1D Fax compression

PDF Security

These options control the security options available in creation of PDF output files. Table values in **bold** text are the default value for that setting.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="Create secure PDF "
  Description ="Create secure PDF, no password">
  <Settings>

    <!-- Output file options -->
    <add Name ="Devmode settings;Color" Value="1"/>
    <add Name ="Save;Output File Format" Value="PDF Multipaged"/>
    <add Name ="Save;Prompt" Value="0"/>
    <add Name ="Save;Overwrite" Value="1"/>
    <add Name ="PDF File Format;PDF Standard" Value="None"/>
    <add Name ="PDF File Format;Use compression" Value="1"/>
    <add Name ="PDF Security;Use Security" Value="1"/>
    <add Name ="PDF Security;Encrypt Level" Value="1"/>
    ...

  </Settings>
</Profile>
```

Conversion Settings - PDF Security	
Name:	PDF Security;Use Security
Values:	0 - No PDF security 1 - Enable PDF security
Name:	PDF Security;Encrypt Level
Values:	Values: 0 - Sets 40-bit encryption level 1 - Sets 128-bit encryption level
Name:	PDF Security;Can Copy
Values:	0 - Do not allow users to copy text and graphics 1 - Allow users to copy text and graphics
Name:	PDF Security;Can Print
Values:	0 - Do not allow users to print the document 1 - Allow users to print the document
Name:	PDF Security;Can Change Doc
Values:	0 - Do not allow users to change the document 1 - Allow users to change the document

Conversion Settings - PDF Security**Name:** PDF Security;Can ChangeOther**Values:** 0 - Do not allow users to add or change comments and form fields
1 - Allow users to add or change comments and form fields**Name:** PDF Security;User Pswd On**Values:** 0 - No user password required to open document
1 - User password required to open document**Name:** PDF Security;User Pswd**Values:** The user password.**Name:** PDF Security;Owner Pswd On**Values:** 0 - No owner password required to change document
1 - Owner password required to change document**Name:** PDF Security;Owner Pswd**Values:** Owner password

JPEG File Format

These options control the compression levels of JPEG files. Table values in **bold** text are the default value for that setting.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="JPEG 300dpi"
  Description ="Create JPEG, compress color as Medium, grey as High.">
  <Settings>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name ="Save;Output File Format" Value="JPEG"/>
    <add Name ="Save;Prompt" Value="0"/>
    <add Name ="Save;Overwrite" Value="1"/>
    <add Name ="Save;Color reduction" Value="Optimal"/>
    <add Name ="JPEG File Format;Color compression" Value="Medium Quality"/>
    <add Name ="JPEG File Format;Greyscale compression" Value="High Quality"/>
    ...

  </Settings>
</Profile>
```

Conversion Settings - JPEG File Format	
Name:	JPEG File Format;Color compression
Values:	High Quality - High quality JPEG compression Medium Quality - Medium quality JPEG compression Low Quality - Low quality JPEG compression
Name:	JPEG File Format;Greyscale compression
Values:	High Quality - High quality JPEG compression Medium Quality - Medium quality JPEG compression Low Quality - Low quality JPEG compression

Processing

These options allow you to do extra processing to the image, such as trimming whitespace, cropping and resampling. Table values in **bold** text are the default value for that setting.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="TIFF 300dpi Serialized with Trim"
  Description="Create TIFF serialized, trim whitespace">
  <Settings>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name="Save;Output File Format" Value="TIFF Serialized"/>
    <add Name="Save;Prompt" Value="0"/>
    <add Name="Processing;Trim Threshold" Value="0"/>
    <add Name="Processing;Trim left" Value="1"/>
    <add Name="Processing;Trim top" Value="1"/>
    <add Name="Processing;Trim bottom" Value="1"/>
    <add Name="Processing;Trim right" Value="1"/>
    ...

  </Settings>
</Profile>
```

Conversion Settings - Processing

Name: Processing;Units

Specifies what unit of measurement is used for settings such as custom paper width or hardware margin. Units can be entered in inches (8.50in) or centimeters (21.59cm), provided the unit designation of inches (in) or centimeters (cm) is given. Also accepted are units entered in as hundredths of an inch (.01 Inches) or tenths of a millimeter(.1 Millimeters)

Values: **.01 Inches**
.1 Millimeters

Name: Processing;Trim left

Trim all areas from the left side of the page, based on the *Trim Threshold* below.

Values: **0** - Do not trim left side of page
1 - Trim left side of page

Name: Processing;Trim top

Trim all areas from the top edge of the page, based on the *Trim Threshold* below.

Values: **0** - Do not trim top of page
1 - Trim top of page

Conversion Settings - Processing**Name: Processing;Trim right**

Trim all areas from the right side of the page, based on the *Trim Threshold* below.

Values: 0 - Do not trim right side of page
1 - Trim right side of page

Name: Processing;Trim bottom

Trim all areas from the bottom edge of the page, based on the *Trim Threshold* below.

Values: 0 - Do not trim bottom of page
1 - Trim bottom of page

Name: Processing;Trim Threshold

All areas on the chosen sides of the image that fall at or below the chosen intensity level, or trim threshold. The intensity level is used to decide what pixels get thrown away. Colors are converted to a grayscale palette, and then compared to the chosen intensity level. Trimming on any side stops as soon as a pixel is encountered that is greater the chosen level. 0 is white, and 100 is black.

Values: 0-100

Name: Processing;Crop

Enable or disable the cropping options.

Values: 0 - Disable cropping
1 - Enable cropping

Name: Processing;Crop Option

Cropping can be specified in either of two ways: as page margins, or as a central area or region on the page.

Values: 0 - Crop region
1 - Crop margins

Name: Processing;Crop left

Applies when Crop Option is set to *crop region*.

Values: 0 - 8000000 - Range in hundredths of an inch
0 - 20000000 - Range in tenths of a millimeter
0.000in - 80000.000in - Range in inches
0.000cm - 200000.000cm - Range in centimeters

Conversion Settings - Processing

Name: **Processing;Crop top**

Applies when Crop Option is set to 0 for *crop region*.

Values: Same as *Processing;Crop left* above

Name: **Processing;Crop width**

Applies when Crop Option is set to 0 for *crop region*.

Values: Same as *Processing;Crop left* above.

Name: **Processing;Crop height**

Applies when Crop Option is set to 0 for *crop region*.

Values: Same as *Processing;Crop left* above

Name: **Processing;Crop margin left**

Applies when Crop Option is set to 1 for *crop margins*.

Values: Same as *Processing;Crop left* above

Name: **Processing;Crop margin top**

Applies when Crop Option is set to 1 for *crop margins*

Values: Same as *Processing;Crop left* above

Name: **Processing;Crop margin right**

Applies when Crop Option is set to 1 for *crop margins*

Values: Same as *Processing;Crop left* above

Name: **Processing;Crop margin bottom**

Applies when Crop Option is set to 1 for *crop margins*

Values: Same as *Processing;Crop left* above

Name: **Processing;Copy**

Enable or disable the copy options. The Copy feature allow you to copy each page of the document to a larger or smaller page.

Values: **0** - Disable copy options
 1 - Enable copy options

Conversion Settings - Processing**Name:** **Processing;Copy to width**

The width of the new image

Values: **0** - 8000000 - Range in hundredths of an inch
0 - 20000000 - Range in tenths of a millimeter
0.000in - 80000.000in - Range in inches
0.000cm - 200000.000cm - Range in centimeters

Name: **Processing;Copy to height**

The height of the new image.

Values: Same as *Processing;Copy to width* above.**Name:** **Processing;Copy to IAM Left**

The desired left area margin settings for the new image.

Values: Same as *Processing;Copy to width* above**Name:** **Processing;Copy to IAM Top**

The desired top area margin settings for the new image.

Values: Same as *Processing;Copy to width* above**Name:** **Processing;Copy to IAM Right**

The desired right area margin settings for the new image.

Values: Same as *Processing;Copy to width* above**Name:** **Processing;Copy to IAM Bottom**

The desired bottom area margin settings for the new image.

Values: Same as *Processing;Copy to width* above**Name:** **Processing;Copy H align**

How to horizontally align the copied image area.

Values: Left - Align the copied image to the left on the page
Middle - Align the copied image horizontally center on the page
Right - Align the copied image to the right of the page

Conversion Settings - Processing

Name: **Processing;Copy V align**

How to vertically align the copied image area.

Values: Top - Align the copied image to the top of the page
Middle - Align the copied image vertically centered on the page
 Bottom - Align the copied image to the bottom of the page

Name: **Processing;Copy Page Scaling**

How to place the original page in the new image.

Values: **0** - Fit to Page
 1 - Actual Size

Name: **Processing;Copy Page Scaling Shrink Larger**

Scales the image down to fit the new image size if the original image is larger.

Values: 0 - Do not shrink page to fit
 1 - Shrink page to fit

Name: **Processing;Copy Page Scaling Lock Aspect Ratio**

Use this option on to prevent distortion when scaling larger or smaller image to different image sizes.

Values: 0 - Do not maintain page aspect ratio when scaling
 1 - Maintain page aspect ratio when scaling

Name: **Processing;Resample**

Scale the output file to a particular width and height in pixels, as a percentage of the original size, or by setting a new image resolution (DPI).

Values: **0** - Disable resampling options
 1 - Enable resampling options

Name: **Processing;Resample Units**

Values: **0** - Pixels
 1 - Percentage
 2 - DPI

Name: **Processing;Resample Lock Aspect Ratio**

Values: 0 - Do not maintain page aspect ratio when resampling
 1 - Maintain page aspect ratio when resampling

Conversion Settings - Processing	
Name:	Processing;Resample Pixels Width Desired width in pixels.
Values:	0-4294967295 pixels, default width is 200 .
Name:	Processing;Resample Pixels Height Desired height in pixels.
Values:	0-4294967295 pixels, default height is 200 .
Name:	Processing;Resample Width Percentage Change the width as a percentage of the original size.
Values:	1 to 500, default is 100 .
Name:	Processing;Resample Height Percentage Change the height as a percentage of the original size.
Values:	1 to 500, default is 100
Name:	Processing;Resample X DPI Change the X resolution of the image.
Values:	50-3600, default is 200
Name:	Processing;Resample Y DPI Change the Y resolution of the image.
Values:	50-3600, default is 200
Name:	Processing;Brightness Adjust Allows you to lighten or darken the images or text on your incoming pages.
Values:	--100 to -1 - darkens the image 0 - no change 1 to 100 - lightens the image
Name:	Processing;Rotate portrait Rotates portrait orientated images the desired degrees counter-clockwise.
Values:	0 , 90, 180, or 270

Conversion Settings - Processing**Name:** **Processing;Rotate landscape**

Rotates landscape orientated images the desired degrees counter-clockwise.

Values: **0, 90, 180, or 270**

Advanced Features

These options allow control of some of the advanced features, such as custom paper size and text extraction. Table values in **bold** text are the default value for that setting.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="TIFF 300dpi Serialized Extract Text"
  Description ="TIFF 300dpi Serialized Extract Text">
  <Settings>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name ="Save;Output File Format" Value="TIFF Serialized"/>
    <add Name ="Save;Prompt" Value="0"/>
    <add Name ="Advanced Features;Extract Text" Value="1"/>
    <add Name ="Advanced Features;Extract Text Layout" Value="Physical"/>
    ...

  </Settings>
</Profile>
```

Conversion Settings - Advanced Features	
Name:	Advanced Features;Units
	Specifies what unit of measurement is used for settings such as custom paper width or hardware margin. Units can be entered in inches (8.50in) or centimeters (21.59cm), provided the unit designation of inches (in) or centimeters (cm) is given. Also accepted are units entered in as hundredths of an inch (.01 Inches) or tenths of a millimeter(.1 Millimeters).
Values:	.01 Inches .1 Millimeters
Name:	Advanced Features;Custom Paper Enable
	Enable or disable custom paper size.
Values:	0 - disable custom paper size 1 - enable custom paper size
Name:	Advanced Features;Custom Paper Width
	Specify the width of the custom paper size. <i>Custom Paper Enable</i> must be 1 for this to be used.
Values:	25 - 8000000 (default 850) - Range in hundredths of an inch 64 - 20000000 - Range in tenths of a millimeter 0.250in - 80000.000in - Range in inches 0.640cm-200000.000cm - Range in centimeters

Conversion Settings - Advanced Features**Name: Advanced Features;Custom Paper Height**

Specify the height of the custom paper size. *Custom Paper Enable* must be 1 for this to be used.

Values: 25 - 8000000 (default **1100**) - Range in hundredths of an inch
 64 - 20000000 - Range in tenths of a millimeter
 0.250in - 80000.000in - Range in inches
 0.640cm-200000.000cm - Range in centimeters

Name: Advanced Features;Hardware Margin Left

Values: 0 - 100 (default = **0**) - Range in hundredths of an inch
 0 - 254 - Range in tenths of a millimeter
 0.000in-1.000in - Range in inches
 0.000cm-2.540cm - Range in centimeters

Name: Advanced Features;Hardware Margin Top

Values: 0 - 100 (default = **0**) - Range in hundredths of an inch
 0 - 254 - Range in tenths of a millimeter
 0.000in-1.000in - Range in inches
 0.000cm-2.540cm - Range in centimeters

Name: Advanced Features;Printer Area Margin Left

Values: 0 - 8000000 (default = **0**) - Range in hundredths of an inch
 0 - 20000000 - Range in tenths of a millimeter
 0.000in - 80000.000in - Range in inches
 0.000cm-200000.000cm - Range in centimeters

Name: Advanced Features;Printer Area Margin Top

Values: 0 - 8000000 (default = **0**) - Range in hundredths of an inch
 0 - 20000000 - Range in tenths of a millimeter
 0.000in - 80000.000in - Range in inches
 0.000cm-200000.000cm - Range in centimeters

Name: Advanced Features;Printer Area Margin Right

Values: 0 - 8000000 (default = **0**) - Range in hundredths of an inch
 0 - 20000000 - Range in tenths of a millimeter
 0.000in - 80000.000in - Range in inches
 0.000cm-200000.000cm - Range in centimeters

Name: Advanced Features;Printer Area Margin Bottom

Values: 0 - 8000000 (default = **0**) - Range in hundredths of an inch
 0 - 20000000 - Range in tenths of a millimeter
 0.000in - 80000.000in - Range in inches
 0.000cm-200000.000cm - Range in centimeters

Conversion Settings - Advanced Features**Name:** **Advanced Features;Extract Text**

Enable this to also create a separate text file containing all of the textual elements of your source document.

Values: **0** - do not extract text
 1 - extract text into a separate text file

Name: **Advanced Features;Extract Text Filepath**

Path to file receiving extracted text.

Values: Full path to file to store text.

Name: **Advanced Features;Extract Text Layout**

Choose the layout of the text file.

Values: **Physical**
 Matches the format of the text in the original file.
 Raw
 Saves the text in the order in which it was sent to the driver. This may not be the same order in the original file.
 None
 No formatting is attempted. All text is written to the file as it is received

Name: **Advanced Features;Extract Text Encoding**

Choose the encoding of the text file.

Values: ANSI
 UTF-8
 UTF-16

Name: **Advanced Features;Extract Text EOL**

Values: **Windows**
 Lines end with the CRLF line feed
 Mac
 Lines end with the LF line feed
 Unix
 Lines end with the CR line feed

Name: **Advanced Features;Extract Text Emit Page Breaks**

Values: **0**
 1

Name: **Advanced Features;Control Strings Enabled**

Values: **0**
 1

Endorsement Options

These options control the behavior of the endorsements that can be stamped on the output created by Document Conversion Service.

Endorsements are the placing of additional header and footer information at the top and bottom of each page. See also [Watermark Stamping](#) to add watermarks to the page content.

Header and footers can contain text such titles and page numbers. The default height of both the header and the footer is 12 points; this can be adjusted individually as needed.

Both the header and footer can be made up of three separate sections - a left section, a center section and a right section. The width of each section can be set individually to allow for text wrapping within each section. The default width for each section is the width of the page. Text in the top left and bottom left section is always left justified, text in the top center and bottom center section is always centered and text in top right and bottom right sections is always right justified.

The data displayed in each part of the header or footer can be formatted using the [Endorsement Formatting Codes](#) to add page number and total page count information to your header and footer text, as well as to display the text in different fonts, font sizes, colors and other text attributes such as bold, italic and underline. The default font used is Arial at 12 points.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="Endorsed TIFF 300dpi"
  Description="Created TIFF with header and footers.">
  <Settings>
    <!-- Add header and footers for each page -->
    <add Name="Endorsements;Enable" Value="1"/>
    <add Name="Endorsements;HeaderHeightInPoints" Value="20"/>

    <!-- Change the text color and formatting. -->
    <add Name="Endorsements;HeaderLeftFormat"
      Value="&#amp;KFF0000&#amp;BInternal Use&#amp;B"/>

    <!-- Multiline. -->
    <add Name="Endorsements;HeaderRightFormat"
      Value="Confidential&#x0A;DO NOT COPY"/>

    <!-- Change text style and size. -->
    <add Name="Endorsements;FooterHeightInPoints" Value="20"/>
    <add Name="Endorsements;FooterCenterFormat"
      Value="&#amp;'Courier'&#amp;P of &#amp;N"/>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name="Save;Output File Format" Value="TIFF Multipaged"/>
    ...
  </Settings>
</Profile>
```

Conversion Settings - Endorsements Header and Footer Options

Name: Endorsements;Enable

Values: 0 - Do not add endorsements
1 - Add specified endorsements to each page

Name: Endorsements;HeaderHeightInPoints

Values: The height of the header area in points. The default is 12 points.

Conversion Settings - Endorsements Header and Footer Options

Name: **Endorsements;HeaderLeftWidthInPoints**

Values: The width of the left section of the header area in points. The default is the width of the page.

Name: **Endorsements;HeaderCenterWidthInPoints**

Values: The width of the center section of the header area in points. The default is the width of the page.

Name: **Endorsements;HeaderRightWidthInPoints**

Values: The width of the right section of the header area in points. The default is the width of the page.

Name: **Endorsements;HeaderLeftFormat**

Values: The text, with [Endorsement Formatting Codes](#) as needed, to put in the left section of the header.

Name: **Endorsements;HeaderCenterFormat**

Values: The text, with [Endorsement Formatting Codes](#) as needed, to put in the center section of the header.

Name: **Endorsements;HeaderRightFormat**

Values: The text, with [Endorsement Formatting Codes](#) as needed, to put in the right section of the header.

Name: **Endorsements;FooterHeightInPoints**

Values: The height of the footer area in points. The default is 12 points.

Name: **Endorsements;FooterLeftWidthInPoints**

Values: The width of the left section of the footer area in points. The default is the width of the page.

Name: **Endorsements;FooterCenterWidthInPoints**

Values: The width of the center section of the footer area in points. The default is the width of the page.

Name: **Endorsements;FooterRightWidthInPoints**

Values: The width of the right section of the footer area in points. The default is the width of the page.

Name: **Endorsements;FooterLeftFormat**

Values: The text, with [Endorsement Formatting Codes](#) as needed, to put in the left section of the footer.

Conversion Settings - Endorsements Header and Footer Options

Name: **Endorsements;FooterCenterFormat**

Values: The text, with [Endorsement Formatting Codes](#) as needed, to put in the center section of the footer.

Name: **Endorsements;FooterRightFormat**

Values: The text, with [Endorsement Formatting Codes](#) as needed, to put in the right section of the header.

Endorsement Formatting Codes

The following formatting codes are used to format the text strings placed in the headers and footers. If you are using the XML profiles to configure the endorsements you will need to use the XML character entities `&` and `"` to represent the ampersand (`&`) and quotation marks (`"`) to allow the XML data to be interpreted correctly.

Header and Footer Formatting Codes		
XML Code	String Code	Description
<code>&amp;P</code>	<code>&P</code>	<p>This code is replaced by the current page number.</p> <p>XML Example:</p> <pre><add Name="Endorsements;HeaderLeftFormat" Value="Page &amp;P" /></pre> <p>String Example:</p> <pre>item.Set("Endorsements;HeaderLeftFormat", "Page &P")</pre>
<code>&amp;N</code>	<code>&N</code>	<p>This code is replaced by the total number of pages in the output file.</p> <p>XML Example:</p> <pre><add Name="Endorsements;HeaderLeftFormat" Value="Page &amp;P of &amp;N" /></pre> <p>String Example:</p> <pre>item.Set("Endorsements;HeaderLeftFormat", "Page &P of &N")</pre>
<code>&amp;B</code>	<code>&B</code>	<p>Turns bold formatting on and off. All text after the first occurrence of the formatting code will be bold until the same formatting code is encountered again.</p> <p>XML Example:</p> <pre><add Name="Endorsements;HeaderLeftFormat" Value="&amp;BInternal Use&amp;B - Confidential" /></pre> <p>String Example:</p> <pre>item.Set("Endorsements;HeaderLeftFormat", "&BInternal Use Only&B - Confidential")</pre>
<code>&amp;I</code>	<code>&I</code>	<p>Turns italic formatting on and off. All text after the first occurrence of the formatting code will be italicized until the same formatting code is encountered again.</p> <p>XML Example:</p> <pre><add Name="Endorsements;HeaderLeftFormat" Value="&amp;IDo Not Copy&amp;I - Confidential" /></pre> <p>String Example:</p> <pre>item.Set("Endorsements;HeaderLeftFormat", "&IDo Not Copy&I - Confidential")</pre>
<code>&amp;U</code>	<code>&U</code>	<p>Turns font underlining on and off. All text after the first occurrence of the formatting code will be underlined until the same formatting code is encountered again.</p> <p>XML Example:</p> <pre><add Name="Endorsements;HeaderLeftFormat" Value="&amp;UDo Not Copy&amp;U - Confidential" /></pre> <p>String Example:</p> <pre>item.Set("Endorsements;HeaderLeftFormat", "&UDO Not Copy&U - Confidential")</pre>

Header and Footer Formatting Codes

XML Code	String Code	Description
&S	&S	<p>Turns font strike through formatting on and off. All text after the first occurrence of the formatting code will be struck through (a line down the middle of the text) until the same formatting code is encountered again.</p> <p>XML Example:</p> <pre><add Name="Endorsements;HeaderLeftFormat" Value="&SInternal Use&S - Confidential"/></pre> <p>String Example:</p> <pre>item.Set("Endorsements;HeaderLeftFormat", "&SInternal Use Only&S - Confidential")</pre>
&X	&X	<p>Turns font superscript formatting on and off. All text after the first occurrence of the formatting code will be printed in superscript (appears smaller than the normal line of type and is set slightly above it) until the same formatting code is encountered again.</p> <p>XML Example:</p> <pre><add Name="Endorsements;HeaderLeftFormat" Value="This is &Xsuperscript text&X - Confidential"/></pre> <p>String Example:</p> <pre>item.Set("Endorsements;HeaderLeftFormat", "This is &Xsuperscript text&X - Confidential")</pre>
&Y	&Y	<p>Turns font subscript formatting on and off. All text after the first occurrence of the formatting code will be printed in subscript (appears smaller than the normal line of type and is set slightly below it) until the same formatting code is encountered again.</p> <p>XML Example:</p> <pre><add Name="Endorsements;HeaderLeftFormat" Value="This is &Ysubscript text&Y - Confidential"/></pre> <p>String Example:</p> <pre>item.Set("Endorsements;HeaderLeftFormat", "This is &Ysubscript text&Y - Confidential")</pre>
&'fontname'	&'fontname'	<p>Sets the font to be used for the following text. All text after the occurrence of the formatting code will be printed in the specified font until another font formatting code is encountered again. The default font is Arial.</p> <p>XML Example:</p> <pre><add Name="Endorsements;HeaderLeftFormat" Value="This is Arial and &'Verdana'this is Verdana."/></pre> <p>String Example:</p> <pre>item.Set("Endorsements;HeaderLeftFormat", "This is Arial and &'Verdana'this is Verdana.")</pre>
&n	&n	<p>Sets the font size, in points, to be used for the following text, where n is replaced with the desired point size. All text after the occurrence of the formatting code will be printed in the specified font size until another font size formatting code is encountered again. The default font size is 12 points.</p> <p>XML Example:</p> <pre><add Name="Endorsements;HeaderLeftFormat" Value="&14This is Arial 14 point."/></pre>

Header and Footer Formatting Codes		
XML Code	String Code	Description
		<p>String Example:</p> <pre>item.Set("Endorsements;HeaderLeftFormat", "&l4This is Arial 14 point.")</pre>
<code>&amp;K000000</code>	<code>&K000000</code>	<p>Changes the color of the text. All text after the occurrence of the formatting code will be printed in the color specified until another color formatting code is encountered again. The default color is Black. The color is specified as six character RGB code.</p> <p>XML Example:</p> <pre><add Name="Endorsements;HeaderLeftFormat" Value="This is &amp;KFF0000Red, this is &amp;K00FF00Green."/></pre> <p>String Example:</p> <pre>item.Set("Endorsements;HeaderLeftFormat", "This is &KFF0000Red, this is &K00FF00Green.")</pre>
<code>&amp;&amp;</code>	<code>&&</code>	<p>Allows the insertion of an ampersand character into the text.</p> <p>XML Example:</p> <pre><add Name="Endorsements;HeaderLeftFormat" Value="Printed by Company &amp;&amp;Company"/></pre> <p>String Example:</p> <pre>item.Set("Endorsements;HeaderLeftFormat", "Printed by Company && Company")</pre>
<code>&#x0A;</code>	<code>\r\n</code>	<p>Allows the insertion of a newline character into the text.</p> <p>XML Example:</p> <pre><add Name="Endorsements;HeaderLeftFormat" Value="Line 1&#x0A;Line 2"/></pre> <p>String Example:</p> <pre>item.Set("Endorsements;HeaderLeftFormat", "Line 1\r\nLine 2.")</pre>

Watermark Stamping

These options allow the placement of a centered, diagonal watermark on each page. The watermark text runs from bottom left to the top right of the page with the outline of each letter being printed. Table values in **bold** text are the default value for that setting.



Sample Profile

```
<?xml version="1.0" encoding="utf-8"?>
<Profile Type="0"
  DisplayName="TIFF 300dpi Serialized Extract Text"
  Description="TIFF 300dpi Serialized Extract Text">
  <Settings>

    <!-- Output file options -->
    <add Name="Devmode settings;Resolution" Value="300"/>
    <add Name="Save;Output File Format" Value="TIFF Serialized"/>
    <add Name="Save;Prompt" Value="0"/>
    <add Name="WatermarkStamp;Enabled" Value="1"/>
    <add Name="WatermarkStamp;CenteredDiagonalText" Value="DRAFT"/>
    <add Name="WatermarkStamp;CenteredDiagonalFontSizeInPoints" Value="36"/>
    ...

  </Settings>
</Profile>
```

Conversion Settings - Advanced Features

Name:	WatermarkStamp;Enabled
	Enable or disable the watermark stamping feature.
Values:	0 - disable watermark stamping 1 - enable watermark stamping
Name:	WatermarkStamp;CenteredDiagonalText
Values:	The text to display as the watermark stamp.
Name:	WatermarkStamp;CenteredDiagonalFontSizeInPoints
Values:	The font size of the watermark text in points. Default is 36.