



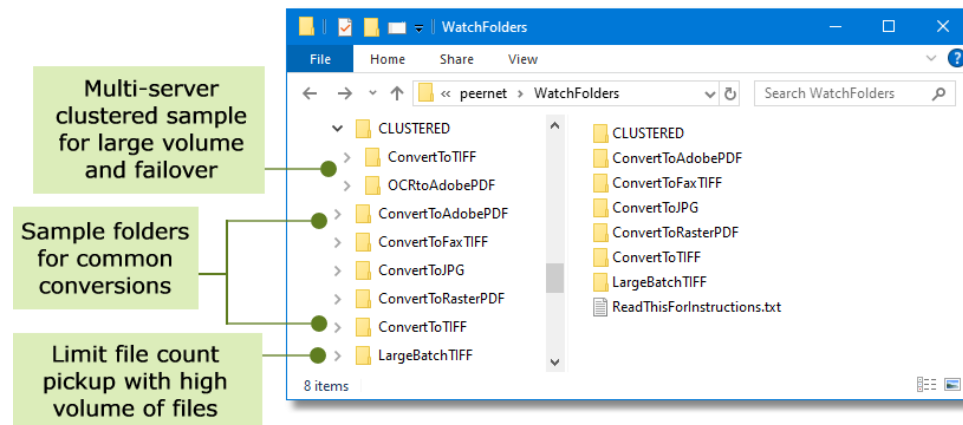
Watch Folder Quick Start Guide

The Watch Folder Service watches one or more *hot* folders at a time and periodically converts any files or folders dropped into those folders. Each folder definition controls what file types are picked up to be converted, the type of file to create and where to save the final file.

This guide covers the basics of changing one of these watch folder definitions to monitor a different folder, set the converted file format and to save the new file into a different location.

It also includes advanced sections on handling large volumes of files, processing archived Outlook messages and running post-processing commands.

The service configuration includes several sample watch folders for common conversion requests.



Each watched folder uses a series of underlying folders to control the pickup, storing, converting and final location of the original and converted files.

These folders can all be on the same computer as the Watch Folder Service but it is more common to use a combination of shared folders on another computer (server) for the *Input*, *Output*, *Failed* and *Completed* folders and have only the internal *Staging* and *Working* folders local to the computer running the service.

Each of these folders needs to be unique for each watch folder definition.

Input Folder

- Files to be converted are dropped into this folder. Often a shared network folder.

Staging Folder

- Move file from Input to here on file pickup. This should be a local folder.

Working Folder

- Stores the new file while it is being created. This should also be a local folder.

Output Folder

- Move the new file here on success. Also often a shared network folder.

Failed Folder

- Copy original file here if conversion fails. Can be a network or local folder.

Completed Folder

- Optionally keep a copy of the original file. Can be a network or local folder.



The Watch Folder Configuration File

The configuration file is an XML-based file that can be edited in the DCS Editor by going to the DCS Dashboard - Watch Folder Settings - Edit Configuration.

Each of the sample watch folders has its own collection of settings in the XML file. In each collection there are two main groups of settings, the Folder settings and the Output File settings.

Folder Settings

The folder settings define the folder being watched, what files to look for and how often to check the folder.

Each watch folder needs separate folder locations for the Input, Staging, Working, Output, and Failed folders. These folder locations also need to be unique for each individual watch folder.

The Completed folder keeps a copy of all files dropped into the Input folder to be converted. It can be set as an empty string if there is no need to keep the files.

Use the SearchFilter to set what file types to look for and the PollingInterval to set the maximum wait period between checking for files to be picked up.

```
<!-- This watch folder creates 300 DPI Optimized TIFF Images -->
<WatchFolder Name="ConvertToTIFF">
  <Settings>
    <!-- Folder options -->
    <add Name="Enabled" Value="true" />
    <add Name="InputFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Input"/>
    <add Name="SearchFilter" Value="*.*/>
    <add Name="IncludeSubFolders" Value="True"/>
    <add Name="DeleteInputSubFolders" Value="true"/>
    <add Name="StagingFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Staging"/>
    <add Name="WorkingFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Working"/>
    <add Name="FailedFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Failed"/>
    <add Name="CompletedFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Completed"/>
    <add Name="OutputFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Output"/>
    <add Name="PollingInterval" Value="15000"/>
  </Settings>
</WatchFolder>
```

Output File Settings

These settings control the type of file being created by the conversion.

Choose searchable or raster PDF files or create images such as TIFF, JPEG, BMP and others. Create a single multipaged PDF or TIFF for each file, or create a separate, serialized file for each page in the file.

Control the color in the final output with automatic color reduction to minimal used colors, grayscale or black and white.

Set the image resolution and compression methods used to control the size and quality of the converted file.

```
<!-- Output file options -->
<add Name="Devmode settings;Resolution" Value="300"/>

<add Name="Save;Output File Format" Value="TIFF Multipaged"/>
<!-- Replace the above with this to create serialized images. -->
<!-- <add Name="Save;Output File Format" Value="TIFF Serialized"/> -->

<add Name="Save;Append" Value="0"/>
<add Name="Save;Color reduction" Value="Optimal"/>
<add Name="Save;Dithering method" Value="Halftone"/>

<!-- This creates file.ext.tif, change to 1 to create file.tif-->
<add Name="Save;Remove filename extension" Value="0" />

<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"/>
```

Additional Settings

Each watch folder also contains additional groups of settings for the following features:

- processing [Outlook Message Archives](#) (*.msg) and all attachments
- set custom commands to run [post-conversion](#) on converted or failed files
- [batch processing](#) to safely handle large volumes of data
- [clustered processing](#) for high throughput and fail over protection with multiple Document Conversion Service installations

PEERNET Document Conversion Service 3.0



Tell Me What Folder to Watch

The quickest way to get started is to modify a sample watch folder that is already similar to what you want to do.

The InputFolder is the *hot* folder, or the location that will be monitored for files to convert. This can be a folder local to the computer running the Watch Folder Service or a network shared folder.

Files are chosen based on their extension using the SearchFilter setting. Multiple file extensions are listed separated by the pipe (|) symbol while a filter of "*. *" will pick up everything.

The folder is checked based on the time set for the PollingInterval; set in milliseconds, the default of 15000 checks the folder every 15 seconds.

```
<!-- Folder options -->
<add Name="Enabled" Value="true" />
<add Name="InputFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Input"/>
<add Name="SearchFilter" Value="*.doc|*.docx|*.pdf"/>
<add Name="IncludeSubFolders" Value="true"/>
<add Name="DeleteInputSubFolders" Value="true"/>
<add Name="StagingFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Staging"/>
<add Name="WorkingFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Working"/>
<add Name="FailedFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Failed"/>
<add Name="CompletedFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Completed"/>
<add Name="OutputFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Output"/>
<add Name="PollingInterval" Value="15000"/>
<add Name="DCOMComputerName" Value="localhost"/>
<add Name="PollingInterval" Value="5000"/>
<add Name="CopyInstructionsFromResources" Value="ReadMe_ConvertToTIFF"/>
```

Check for files every 5 seconds

Set input on a network share and only convert Word and PDF



When using network share folders keep in mind that the logon account for the Watch Folder Service needs to have full access to the shared folders. This account would have been set up when installing Document Conversion Service but can be changed through the Services control panel as needed.

Tell Me Where to Save the Files

The OutputFolder is where the converted files will be saved. As with the InputFolder, this can be a local folder or a shared folder on another computer.

```
<!-- Folder options -->
<add Name="Enabled" Value="true" />
<add Name="InputFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Input"/>
<add Name="SearchFilter" Value="*.doc|*.docx|*.pdf"/>
<add Name="IncludeSubFolders" Value="true"/>
<add Name="DeleteInputSubFolders" Value="true"/>
<add Name="StagingFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Staging"/>
<add Name="WorkingFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Working"/>
<add Name="FailedFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Failed"/>
<add Name="CompletedFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Completed"/>
<add Name="OutputFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Output"/>
<add Name="PollingInterval" Value="15000"/>
<add Name="DCOMComputerName" Value="localhost"/>
<add Name="PollingInterval" Value="5000"/>
<add Name="CopyInstructionsFromResources" Value="ReadMe_ConvertToTIFF"/>
```

Leave Staging and Working as local folders

Set Output on a network share

Both the StagingFolder and WorkingFolder should be set to local folders on the computer to reduce network traffic, but can also be network shares if needed. These folders are temporary *scratch* folders used internally by the Watch Folder Service.

Files and folders found in the InputFolder are first moved to the StagingFolder to be converted, while the converted files are first stored in the WorkingFolder before being moved to the OutputFolder.

When processing a folder of files dropped into the InputFolder, and subfolders are being processed as well, any underlying folder structure of the dropped folder is maintained throughout the entire conversion process.



Keeping the Original File

The CompletedFolder will store a copy of each file dropped into the InputFolder for conversion. This can be a local folder or a shared folder.

If you do not need to keep the original files, set the CompletedFolder to an empty string to disable this behavior. Take note, however, that this causes all files dropped into the InputFolder to be deleted after conversion.

If you do need to keep them, all files that are successfully converted are copied into a sub folder under the CompletedFolder. The sub folder is named using the date and time the files were picked up from the InputFolder.

The setting UseTimeDateSubFoldersInCompletedFolder can be set to false to have the source files copied directly into the CompletedFolder with no sub folder. Files of the same name will be overwritten.

```
<add Name="StagingFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Staging"/>
<add Name="WorkingFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Working"/>
<add Name="FailedFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Failed"/>
<add Name="CompletedFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Completed"/>
<add Name="OutputFolder" Value="\\ServerA\Data\ConvertToTIFF\Output"/>
<add Name="PollingInterval" Value="10" />
<add Name="DCOMComputerName" Value="localhost"/>
<add Name="TestMode" Value="false"/>
<add Name="NormalizeFileNames" Value="false"/>
<add Name="CopyInstructionsFromResources" Value="ReadMe_ConvertToTIFF"/>

<!-- 0 means no limit -->
<add Name="Polling.MaxFilesToProcessAtATime" Value="0" />
<add Name="Polling.SynchronousFilePickup" Value="false" />

<add Name="UseTimeDateSubFoldersInCompletedFolder" Value="true" />
<add Name="UseTimeDateSubFoldersInFailedFolder" Value="true" />
<add Name="UseCompressedDateTimeFormat" Value="false" />
```

Use dated sub folders
when keeping the
original files

Set as empty string to
delete input files after
conversion

What Happens When a File Fails?

If a file fails to convert it is stored in the folder set as the FailedFolder. This can be a local folder or a shared folder.

Like the CompletedFolder, failed files are normally copied into a sub folder under this folder. This sub folder is also named using the date and time the files were picked up from the InputFolder.

To copy the failed files directly into the folder instead, the UseTimeDateSubFoldersInFailedFolder setting can be set to false. Any files of the same name that already exists in the folder will be overwritten.

```
<add Name="StagingFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Staging"/>
<add Name="WorkingFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Working"/>
<add Name="FailedFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Failed"/>
<add Name="CompletedFolder" Value="" />
<add Name="OutputFolder" Value="C:\PEERNET\WatchFolders\ConvertToTIFF\Output"/>
<add Name="PollingInterval" Value="10" />
<add Name="DCOMComputerName" Value="localhost"/>
<add Name="TestMode" Value="false"/>
<add Name="NormalizeFileNames" Value="false"/>
<add Name="CopyInstructionsFromResources" Value="ReadMe_ConvertToTIFF"/>

<!-- 0 means no limit -->
<add Name="Polling.MaxFilesToProcessAtATime" Value="0" />
<add Name="Polling.SynchronousFilePickup" Value="false" />

<add Name="UseTimeDateSubFoldersInCompletedFolder" Value="true" />
<add Name="UseTimeDateSubFoldersInFailedFolder" Value="true" />
<add Name="UseCompressedDateTimeFormat" Value="false" />
```

Set Failed on a
network share

Use dated sub folders
when storing any
failed files

PEERNET Document Conversion Service 3.0



Set Up Your Output File

The output file settings are at the bottom of each watch folder section.

The Output File Format and DPI

The resolution in *dots per inch* (DPI) of the converted file is controlled by the Devmode settings;Resolution setting. The default of 300 is a good place to start.

The setting Save;Output File Format controls if the converted file will be a [PDF, TIFF, JPEG or other image format](#). This will not need to be changed if the chosen watch folder being edited already creates the type of file you want.

Common Output Types

The following are the most commonly used output formats:

- TIFF Multipaged, TIFF Serialized
- Adobe PDF Multipaged, Adobe PDF Serialized
- OCR scanned PDF and images to searchable PDF
- JPEG

When creating TIFF and PDF documents you can choose between *serialized* (separated) and *multi-paged* image and PDF files. Serialized output saves each page as a separate file while multi-paged output is a single file containing all the pages from the original file.

Controls output file type created and image resolution

```
<!-- Output file options -->
<add Name="Devmode settings;Resolution" Value="300"/>

<add Name="Save;Output File Format" Value="TIFF Multipaged"/>
<!-- Replace the above with this to create serialized images. -->
<!-- <add Name="Save;Output File Format" Value="TIFF Serialized"/> -->

<add Name="Save;Append" Value="0"/>
<add Name="Save;Color reduction" Value="Optimal"/>
<add Name="Save;Dithering method" Value="Halftone"/>

<!-- This creates file.ext.tif, change to 1 to create file.tif-->
<add Name="Save;Remove filename extension" Value="0" />
```

The Output File Name

Each watch folder normally keeps the source file extension as part of the name of the new file. The setting Save;Remove filename extension can be set to 1 to change this.

```
<add Name="Save;Append" Value="0"/>
<add Name="Save;Color reduction" Value="Optimal"/>
<add Name="Save;Dithering method" Value="Halftone"/>

<!-- This creates file.ext.tif, change to 1 to create file.tif-->
<add Name="Save;Remove filename extension" Value="0" />

<add Name="TIFF File Format;BW compression" Value="Group4"/>
<add Name="Save;Append" Value="0"/>
<add Name="Save;Color reduction" Value="Optimal"/>
<add Name="Save;Dithering method" Value="Halftone"/>
<!-- This creates file.ext.tif, change to 1 to create file.tif-->
<add Name="Save;Remove filename extension" Value="1" />
```

Don't use source file extension in the new file name

Controlling the Color in the New File

The Save;Color reduction setting controls if any color is kept in the file, or if the colors are changed to black and white. Other [color reduction](#) options are also available.



Optimal reduces each page to the lowest color count needed. This default is a good place to start.



BW will reduce the document to black and white. This setting is good if you need fax format TIFF.

```
<add Name="Save;Append" Value="0"/>
<add Name="Save;Color reduction" Value="Optimal"/>
<add Name="Save;Dithering method" Value="Halftone"/>

<!-- This creates file.ext.tif, change to 1 to create file.tif-->
<add Name="Save;Remove filename extension" Value="1" />

<add Name="Save;Append" Value="0"/>
<add Name="Save;Color reduction" Value="BW"/>
<add Name="Save;Remove filename extension" Value="1" />
```

Reduce file to black and white

PEERNET Document Conversion Service 3.0



Starting the Service



Before you begin! Document Conversion Service needs to be running to convert files. The Watch Folder Service can be started but will not pick up any files to convert until Document Conversion Service is running and ready to convert files.

Using the Logging Console


The logging console displays live logging of the Watch Folder Service when it is running. It can also open saved log files. The console is opened from the DCS Dashboard by going to Watch Folder Settings - View Logging Console.

It is not meant to run continuously but for monitoring service start up and for intermediate troubleshooting; do not leave the console open for extended periods of time.

Start the Watch Folder Service

With Document Conversion Service started and, optionally, the logging console open, the Watch Folder Service can now be started. Open the DCS Dashboard. At the top you have easy access to starting and stopping both Document Conversion Service and Watch Folder Service.

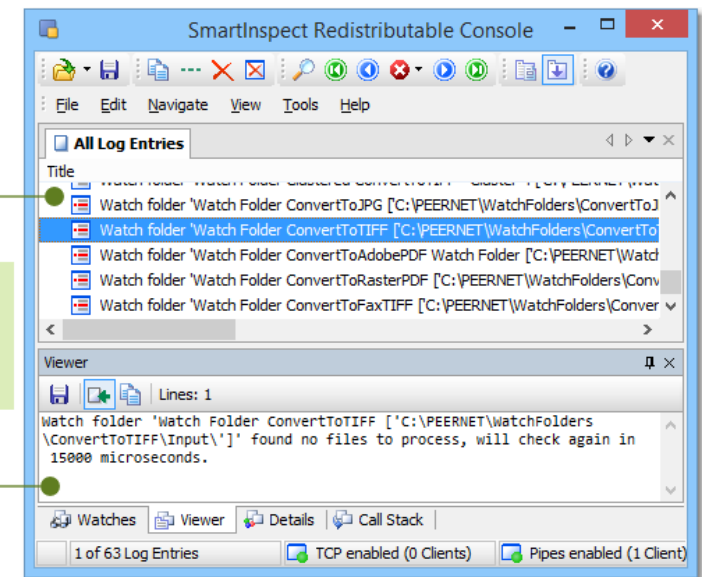
DCS Service:  Running **Watch Folder Service:   Stopped**

Click the green play icon (). The status will change to Starting..., and move to Running once the service is running.

Logging messages will begin to appear in the All Log Entries panel of the logging console.

Messages about each of the monitored folders and any files found to convert as well as any problems with security and permissions on the monitored folders are displayed here.

This information is also saved to a series of log files on disk split by size and date that can be opened in the console for troubleshooting past issues.



Select logging messages to see full information in the viewer



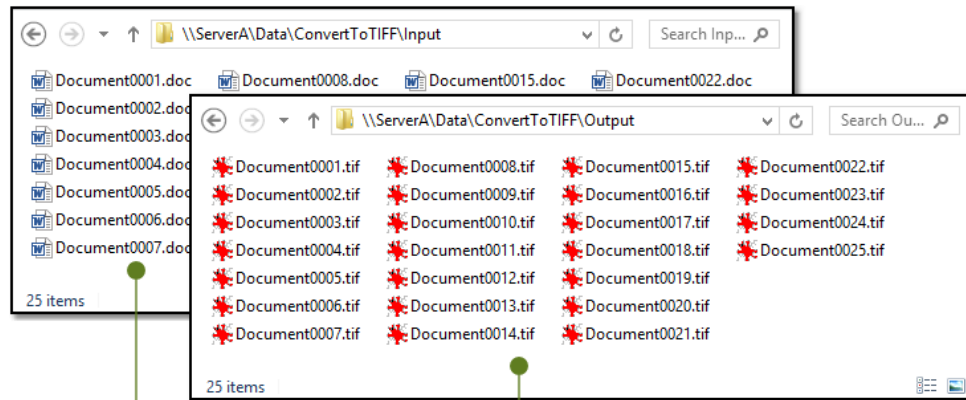
If no messages appear in the logging console then the Watch Folder Service failed to start. This can happen when changes to the configuration file break the file so that it cannot be read. Common causes are quotes (") or angle brackets(<, >) accidentally deleted during editing.

PEERNET Document Conversion Service 3.0



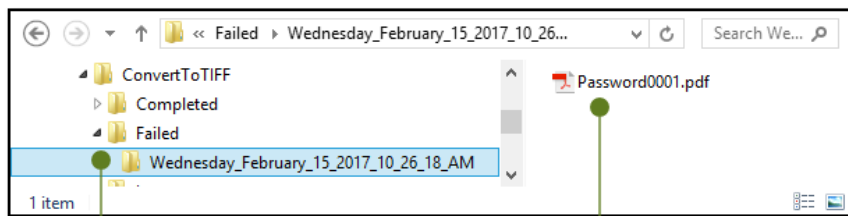
Drop a File to Convert It

Now that the Watch Folder Service is running, files and folders can be dropped into the InputFolder for conversion. Each time the polling interval expires, the files will be picked up, sent to Document Conversion Service to be converted and saved to the OutputFolder.



Drop files in Input to convert and save in Output.

If any files fail to convert they are copied into the FailedFolder. Commonly these files are password protected or the application that Document Conversion Service uses to convert the file is not installed.



Original file saved in FailedFolder under date time subfolder.

Stop the Service

When making any changes to the configuration of the Watch Folder Service, the service will need to be stopped and then restarted to pick up the modifications.

To stop the service, go to the the DCS Dashboard. At the top you have easy access to starting and stopping both Document Conversion Service and Watch Folder Service.

DCS Service: Running

Watch Folder Service: Running

Click the green stop icon (). The status will change to Stopping..., and go to Stopped when the service has stopped running.



If Document Conversion Service needs to be stopped or restarted while Watch Folder Service is busy processing files, it is best to stop the watch folder service first, and then stop the conversion service. Doing this will copy any unprocessed files already queued up in the Watch Folder Service back to the input folder to be picked up on restart.

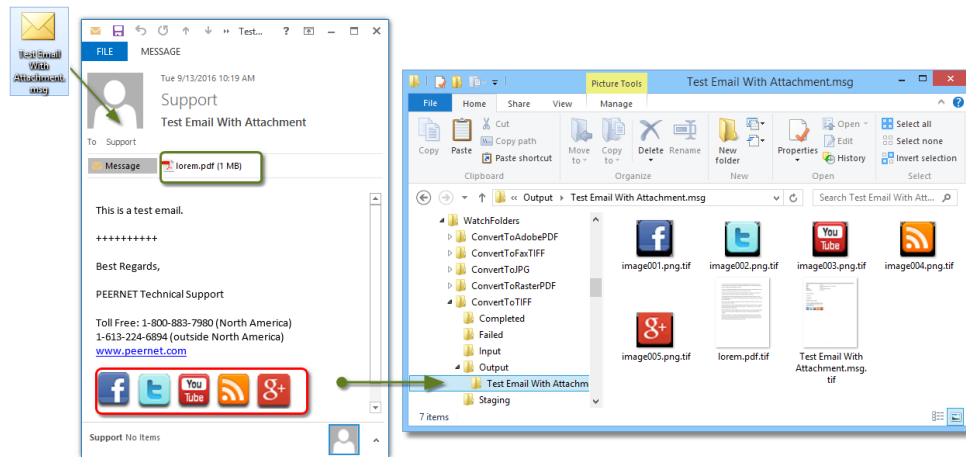
PEERNET Document Conversion Service 3.0



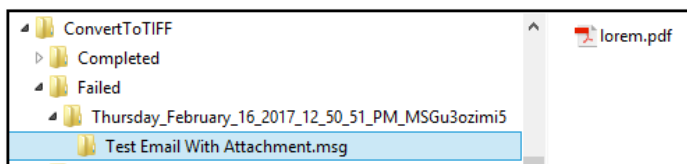
Extract EML and MSG Files and Attachments

Watch Folder Service can extract and convert the message and any attachments and embedded images in Outlook Message files (*.msg, DCS version 3.0.009) and Electronic Mail messages (*.eml, DCS version 3.0.029). This includes recursively processing attachments that are MSG or EML files themselves.

The name of the original MSG or EML file is used to create a new folder under the OutputFolder that contains the resulting collection of converted files from the message. Any files that end up with the same name are automatically numbered to create unique names.



If the main e-mail message or any attachments cannot be converted they are placed in the FailedFolder under a dated sub folder, again in a new folder named using the original MSG file name.



Each sample watch folder includes a group of settings to enable and control how the email and attachments are processed. The settings apply to both MSG and EML messages.

Message attachment processing is initially disabled; remove the comment markers from the setting PreprocessArchiveFormatsFilter to enable it. For both MSG and EML handling, add the filter "*.msg|*.eml".

```
<!-- Preprocess Archive Settings -->
<!-- Comment out this line or set this as empty string to disable
MSG archive processing.-->
<!-- <add Name="PreprocessArchiveFormatsFilter" Value="*.msg" /> -->
<!-- Setting this to false is not recommend as it increases the
<add Name="PreprocessArchiveFormatsFilter" Value="*.msg" />
```

There are settings for filtering which attachments will be converted. Attachments can be separate files or inline in the email. Inline attachments are often pictures used in email signatures but can also be valid documents. Set PreprocessArchive.MSG.IncludeInlineAttachments to *false* to process only external files.

All attachments, inline and external, can be included or excluded using a pipe (|) separated list of file extensions in the PreprocessArchive.MSG.AttachmentsIncludeFilter and the PreprocessArchive.MSG.AttachmentsExcludeFilter settings.

```
<!-- Preprocess MSG Archive Settings -->
<add Name="PreprocessArchive.MSG.IncludeInlineAttachments" Value="true" />
<!-- Pipe (|) separated list of file extensions (e.g *.doc|*.docx) to include. -->
<add Name="PreprocessArchive.MSG.AttachmentsIncludeFilter" Value="" />
<!-- Pipe (|) separated list of file extensions (e.g *.png|*.jpg) to exclude. -->
<add Name="PreprocessArchive.MSG.AttachmentsExcludeFilter" Value="*.doc|*.docx|*.pdf"/>
```

Convert only Word and PDF attachments



See the how to guide [Extracting and Converting Outlook Message Attachments](#) for detailed information on this feature, including unique file naming and using the email Message ID in the name.

Run Post-Processing Commands

The ability to run a command after the file has been converted, or failed to convert, was added to Watch Folder Service in Document Conversion Service 3.0.010.

Each watch folder includes two groups of settings, one to set a command to run when the file is successfully converted, and one to run when a file fails to convert. Each command can be enabled and configured separately.

Post-conversion commands normally do not wait for the command to complete so the Watch Folder Service can quickly move on to the next file. There are options to wait for the command to complete if needed. When waiting for the post-conversion commands to complete there is a maximum timeout value that is always applied to allow the service to move ahead if the command does something unexpected.

Built-in variables are provided to use in when passing in file name information as command parameters.

Success Variables

- \$(SourceFileName) - The file name of the original file.
- \$(OutputFilePath) - The full path to the new converted file.
- \$(OutputFileList) - A Unicode a text file listing all files created.
- (OutputFileNumber) - The index of the current file, starts at 1.
- \$(OutputFileNumberCount) - The number of files created

Failed Variables

- \$(SourceFileName) - The file name of the original file.
- \$(FailedFilePath) - The path to the original file in the FailedFolder.

PEERNET Document Conversion Service 3.0



Shown below is a sample post-conversion setup using the fictional command line tools *DocArchiveUpload.exe* and *TriggerEmailFail.exe*. These executable names are placeholders that you would replace with your own command line tools for your workflow.

Setting a Command for Successful Conversions

```
<!-- Run Command at End On Success -->
<!-- Any command entered here will run ONCE at the end of successful conversion, when all of the files have been created. -->
<!-- Use &quot; to put the command in quotes if there are spaces, and to enclose parameters. -->
<!-- You can set RunAtEnd.Success.RunForEachFile to true to have the command will run ONCE for each file created; -->
<!-- this means the command will run multiple times, once for each file created when creating for serialized output. -->
<!-- Set RunAtEnd.Success.Enabled to true -->
<add Name="RunAtEnd.Success.Enabled" Value="true" />
<!-- Use &quot; to put the command in quotes if there are spaces, and to enclose parameters. -->
<add Name="RunAtEnd.Success.RunForEachFile" Value="false" />
<!-- Use &quot; to put the command in quotes if there are spaces, and to enclose parameters. -->
<add Name="RunAtEnd.Success.CreateOutputFileList" Value="false" />
<!-- Macro for RunAtEnd.Success.Command -->
<add Name="RunAtEnd.Success.Command" Value="&quot;C:\My Tools\DocArchive.exe&quot;" />
<!-- Macro for RunAtEnd.Success.Parameters -->
<add Name="RunAtEnd.Success.Parameters" Value="&quot;$(OutputFilePath)&quot; &quot;$(SourceFileName)&quot;" />
<!-- $(SourceFileName) can also be passed as a parameter to allow correlating the source and output. -->
<!-- $(OutputFileNumber) is the numerical index of the file being processed in the run command. Index starts at 1. -->
<!-- $(OutputFileCount) is the number of files created, not including any text extraction files. -->
<add Name="RunAtEnd.Success.Enabled" Value="false" />
<add Name="RunAtEnd.Success.RunForEachFile" Value="false" />
<add Name="RunAtEnd.Success.CreateOutputFileList" Value="false" />
<add Name="RunAtEnd.Success.Command" Value="" />
<add Name="RunAtEnd.Success.Parameters" Value="&quot;$(OutputFilePath)&quot; &quot;$(SourceFileName)&quot;" />
<add Name="RunAtEnd.Success.StartDirectory" Value="" />
```

Setting a Command for Failed Conversions

```
<!-- Run Command at End On Failure -->
<!-- Any command entered here will run on a failed conversion, on the original source file. -->
<!-- Use &quot; to put the command in quotes if there are spaces, and to enclose parameters. -->
<!-- $(FailedFilePath) is the path to the file in its failed location. -->
<!-- $(SourceFileName) can also be passed as a parameter to allow correlating the source and output. -->
<add Name="RunAtEnd.Fail.Enabled" Value="false" />
<add Name="RunAtEnd.Fail.Command" Value="" />
<add Name="RunAtEnd.Fail.Parameters" Value="&quot;$(FailedFilePath)&quot;" />
<add Name="RunAtEnd.Fail.StartDirectory" Value="" />
<!-- Set RunAtEnd.Fail.Enabled to true -->
<add Name="RunAtEnd.Fail.Enabled" Value="true" />
<add Name="RunAtEnd.Fail.Command" Value="C:\MyTools\TriggerFailEmail.exe" />
<add Name="RunAtEnd.Fail.Parameters" Value="&quot;$(FailedFilePath)&quot; &quot;$(SourceFileName)&quot;" />
<add Name="RunAtEnd.Fail.WaitModeMaxTime" Value="180000" />
```

See the how to guide [Running Post-Process Commands with Watch Folder Service](#) for detailed information on this feature.

Convert a Large Number of Documents

The Watch Folder Service is designed to monitor *hot* folders and convert small groups of files dropped periodically into the folders. When files are found, the entire collection is picked up at once and copied to a staging location to be converted.

To safely convert a very large collection of files, such as a folder containing thousands of files or more, the number of files picked up each time needs to be controlled so as not to overload the system resources or run out of disk space copying files.

Beginning with the Document Conversion Service 3.0.010 the Watch Folder Service supports clustered conversion.

Normally used with Document Conversion Service installed on more than one computer working together on the same group of files, this approach will also work on a single machine to control pickup and processing of very large collections of files.

The sample watch folder configuration *Clustered ConvertToTIFF* has clustering already enabled and will initially pick up four (4) files to convert. As soon as any one of those files has been converted, another file is picked up until all files are converted. This ensures that files are constantly being converted while still keeping system resource use in check.

```
<!-- Clustered Processing -->
<!-- This forces batch mode processing with synchronous wait and -->
<!-- no date time stamp used in the Failed\Completed folders -->
<add Name="ClusteredProcessing.Enabled" Value="true" />
<!-- Override this for clustering to customize pickup -->
<!-- <add Name="ClusteredProcessing.MaxFilesToPickup" Value="4"/> -->
```

Clustering can be enabled for any of the other sample watch folders provided by changing the ClusteredProcessing.Enabled setting to *true*.



Tips for Large Batch Conversion

Don't work from the original source files when converting. We recommend making a copy of the original source files and processing off of the copy.

Another sample watch folder configuration, `LargeBatchTIFF`, can also be used as a starting point for converting large volumes of files.

Unlike the clustered approach, this watch folder will pick up a configurable number of files on every polling interval and send each of those files to be converted. It then waits until each file in the batch has been converted before picking up a new group of files to convert.

While both approaches work, this method can be less efficient with higher license levels of Document Conversion Service as conversion threads are left idle while the last few files in the batch are completed.

The setting `Polling.MaxFilesToProcessAtATime` controls how many files are picked up each time the folder is checked.

`Polling.SynchronousFilePickup` controls if the first batch of files needs to complete before the next group is picked up and is normally set to `true` to control the flow of files through the conversion service.

```
<!-- These settings control the how many files in the batch -->
<!-- are picked up each time, 0 means no limit. -->
<add Name="Polling.MaxFilesToProcessAtATime" Value="10" />
<add Name="Polling.SynchronousFilePickup" Value="true" />
```

Enabling any of the *ClusteredProcessing* settings for a watch folder will override any *Polling* settings that have been set.

Get Other Quick Start Guides

Document Conversion Service Quick Start Guide

A short and quick guide to help you get the conversion service running and converting documents on your server. The [Document Conversion Service Quick Start Guide](#) covers starting the conversion service for the first time, using the logging console and converting documents.

Clustered Watch Folder for Large Data

For larger companies with a need to convert an existing, very large volume of data, or a large volume of files daily, multiple installations of Document Conversion Service and the included Watch Folder service can provide high-performance clustering and fail over protection with load-balancing and 100% conversion stability. The [Clustered Watch Folder Conversion Quick Start](#) shows how easy it is to set up.

Client Server Conversion Quick Start Guide

See how easy it is to setup client-server, or remote, conversion with Document Conversion Service. Remote conversion allows one or more client applications on separate computers to convert files by 'talking' to a server computer, where the actual file conversion takes place. Read through the [Client Server Conversion Quick Start Guide](#) to see how this can work for your company.



Command Line Tools Guide

For Java programmers and adding conversion into scheduled tasks, batch files, or any program that can call an external program, command line tools for converting files and folders are included. If this is what you are looking for, the [Command Line Tools Quick Start Guide](#) will get you started.

.NET API Utility Quick Start Guide

With the included .NET library you can add document conversion directly into your own programs with as little as a single line of code. Have a look at the sample programs provided or follow the steps in the [.NET API Utility Quick Start Guide](#).

Our Support Team is Here to Help

Not sure where to go from here? Have questions? Our support team is always available to help you choose the correct solution for your conversion needs. Please [contact us](#) and we'll be in touch via e-mail, telephone or web meeting to help answer your questions.