

Version
3.0

PNDocConvQueueServiceLib

Programmer's Reference

PEERNET Inc.

Copyright © 2011 - 2017

Updated: 8/2/2017

Table of Contents

Introduction	1
Legal Notices.....	2
Requirements and Prerequisites.....	3
Converters	4
Conversion Settings	8
Creating and Customizing Profiles.....	12
General Converter Options.....	17
Endorsement Options.....	19
Endorsement Formatting Codes.....	23
Word Converter Options.....	27
Excel Converter Options.....	37
PowerPoint Converter Options.....	57
Adobe Reader Options.....	62
Internet Explorer Options.....	65
Ghostscript Converter Options.....	72
Image Converter Options.....	74
OutsideIn AX Options.....	78
Save.....	81
Devmode settings.....	85
Advanced File Naming.....	88
Image Options.....	94
TIFF File Format.....	98
PDF File Format.....	100
PDF Security.....	103
JPEG File Format.....	106
Processing.....	108
Advanced Features.....	116
PNDocConvQueueServiceLib Object Reference	120

- IPNDocConvQueueItem 121
 - Methods 124
 - Contains 124
 - Convert 124
 - Get 124
 - Remove 124
 - Set 124
 - WaitForAvailableConverterPlugIn 124
 - WaitForRunningDocumentConversionService 124
 - Properties 133
 - Errors 133
 - Files 133
 - Images 133
 - Jobs 133
 - Messages 133
 - Pages 133
 - Events 142
 - OnCloseFile 142
 - OnEndConversion 142
 - OnEndImage 142
 - OnEndJob 142
 - OnEndPage 142
 - OnGetNextOutputFileName 142
 - OnLogMessage 142
 - OnOpenFile 142
 - OnReportError 142
 - OnStartConversion 142
 - OnStartImage 142
 - OnStartJob 142
 - OnStartPage 142
- IPNDocConvQueueItemJobs Collection 155
 - Methods 156
 - Item 156
 - Properties 157
 - Count 157
- IPNDocConvQueueItemJob 158
 - Properties 160
 - BytesPrinted 160
 - BytesSpooled 160

Document	160
Files	160
JobGUID	160
JobID	160
Pages	160
PagesPrinted	160
PagesSpooled	160
Status	160
StatusMessage	160
SubmittedTime	160
UserName	160
IPNDocConvQueueItemFiles Collection.....	173
Methods	174
Item	174
Properties	175
Count	175
IPNDocConvQueueItemFile.....	176
Properties	177
Filename	177
Images	177
Job	177
Pages	177
IPNDocConvQueueItemPages Collection.....	181
Methods	182
Item	182
Properties	183
Count	183
IPNDocConvQueueItemPage.....	184
Properties	185
BitsPerPixel	185
Files	185
HeightInPixels	185
Images	185
Job	185
Orientation	185
PageNumber	185
Skipped	185
WidthInPixels	185

XPixelsPerInch	185
YPixelsPerInch	185
IPNDocConvQueueItemImages Collection	196
Methods	197
Item	197
Properties	198
Count	198
IPNDocConvQueueItemImage	199
Properties	200
BitsPerPixel	200
File	200
HeightInPixels	200
Job	200
Orientation	200
PageNumber	200
Pages	200
RotationInDegrees	200
WidthInPixels	200
XPixelsPerInch	200
YPixelsPerInch	200
IPNDocConvQueueItemMessages Collection	211
Methods	212
Item	212
Properties	213
Count	213
IPNDocConvQueueItemMessage	214
Properties	215
Value	215
IPNDocConvQueueItemErrors Collection	216
Methods	217
Item	217
Properties	218
Count	218
IPNDocConvQueueItemError	219
Properties	220
Value	220

Introduction

The PNDocConvQueueServiceLib COM object was created to make it easier for you, the developer, to integrate the conversion of documents using Document Conversion Service into your own applications.

The PNDocConvQueueServiceLib COM object provides the ability to convert documents to various image formats such as TIFF, JPEG, Adobe® PDF, PNG and others.

All of the conversion preferences can be controlled through the PNDocConvQueueServiceLib interface and the conversion is as simple as a single call. When the conversion is complete the call will return with detailed information about the output files created.

Legal Notices

Copyright © 2011 - 2017 by PEERNET Inc. All rights reserved.

PEERNET is a registered trademark of PEERNET Incorporated. Microsoft and Windows are registered trademarks of Microsoft Corporation. All other trademarks and registered trademarks are the properties of their respective holders.

PEERNET Inc.
18 Deakin Street
Suite 208
Ottawa, Ontario
K2E 8B7

Information in this document is accurate up to the time of publication, but does not necessarily reflect enhancements made to PEERNET Inc.'s products, which are released without notice. The software described in this document is furnished under a license agreement. It is against the law to copy the software onto any medium, or to use the software for any purpose, except as specifically allowed in the license agreement. No part of this help system may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose other than the licensed operator's personal use, without the express written permission of PEERNET Inc.

Requirements and Prerequisites

Required Files for Distribution

If your application is running on the same computer as Document Conversion Service you do not need to include any additional files with your application, If the application is running on a different computer, then the Document Conversion Service Client Redistributable needs to be installed or Document Conversion Service needs to be installed and licensed.

Code samples are provided in both VB.NET and C#.NET.

Supported Development Environments

PNDocConvQueueServiceLib can be used in any development environment that can communicate with COM objects. The PNDocConvQueueServiceLib COM object can be used in, but is not limited to, the development environments below:

- Visual Basic .NET 2002, 2003, 2005, 2008, 2010, 2012, 2013
- Visual C# .NET 2002, 2003, 2005, 2008, 2010, 2012, 2013
- Visual C++ 2002, 2003, 2005, 2008, 2010, 2012, 2013
- Delphi
- PowerShell
- VBA Script

Supported Platforms

Only 64-bit operating systems are supported. A minimum of 4GB of memory (RAM) is recommend for best performance.

- Microsoft® Windows Server 2016
- Microsoft® Windows Server 2012 R2
- Microsoft® Windows Server 2012
- Microsoft® Windows Server 2008 R2
- Microsoft® Windows 10
- Microsoft® Windows 8, 8.1
- Microsoft® Windows 7

Converters

PNDocConvQueueServiceLib uses the converters included with Document Conversion Service to provide conversion of the most commonly used document formats. Most of the included converters use the original document's native application to help it do the conversion. This means that the necessary third-party applications also need to be installed and licensed for the converters you want to use.


Each converter has a unique name and it is this name that needs to be passed down to the [Convert](#) method, along with the source file path, to tell Document Conversion Service to use that converter when converting the file. The converter name can be the name of a single converter or a semi-colon separated list of the converter names to try. The first converter in the list that is running is the converter that will be used to convert the document. For example, to convert a PDF file you can use Adobe Acrobat Reader, Ghostscript or the Outside-In AX converter.

Code Sample - C#

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

try
{
    // Create the conversion item
    item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

    ....
    // convert the file, using first matching converter
    item.Convert("Adobe Acrobat Reader;Ghostscript;Outside-In AX",
                @"C:\Test\Report.pdf",
                @"C:\Test\Out\ConvertedReport");
}
catch (Exception ex)
{
    MessageBox.Show(this, "An error has occurred.\n\n" + ex.ToString());
}
finally
{
    if (item != null)
    {
        System.Runtime.InteropServices.Marshal.FinalReleaseComObject(item);
    }
    item = null;
}
```

 **Code Sample - VB**

```

Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

Try
    ' Create the conversion item
    item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

    ....

    ' convert the file, using first matching converter
    item.Convert("Adobe Acrobat Reader;Ghostscript;Outside-In AX", _
                "C:\Test\Report.pdf", _
                "C:\Test\Out\ConvertedReport")

Catch ex As Exception
    MessageBox.Show(Me, "An error has occurred." & vbCrLf & vbCrLf & ex.ToString())

Finally
    If IsNothing(item) Then
        System.Runtime.InteropServices.Marshal.FinalReleaseComObject(item)
        item = Nothing
    End If
End Try

```

The table below outlines the file types that can be converted and their required application, if needed.

Supported Document Type and Converter Name	Third-Party Application
Converter Name: Adobe Acrobat Reader <ul style="list-style-type: none"> • Adobe PDF Documents (*.pdf) 	Adobe Reader X, XI, DC
Converter Name: Autodesk Design Review <ul style="list-style-type: none"> • Design Review Drawings (*.dwf) 	Autodesk Design Review 2012-2013
Converter Name: Autodesk Design Review <ul style="list-style-type: none"> • AutoCAD Drawings (*.dwg) 	Autodesk Design Review 2012-2013 with DWG TrueView 2012-2013 also installed.
Converter Name: Microsoft Excel <ul style="list-style-type: none"> • Excel Workbooks (*.xlsx, *.xlsm, *.xls) • Excel Templates (*.xlt, *.xltn, *.xlt) • Excel Binary Workbook (*.xlsb) 	Microsoft Office 2003 SP3 <i>(with Microsoft Office Compatibility Pack)</i> Microsoft Office 2007 Microsoft Office 2010 (32-bit and 64-bit) Microsoft Office 2013 (32-bit and 64-bit) Microsoft Office 2016 (32-bit and 64-bit)
Converter Name: Ghostscript <ul style="list-style-type: none"> • Postscript Files (*.ps) • Encapsulated Postscript Files (.eps) • Adobe PDF Documents (*.pdf) 	Ghostscript 9.05 or later <i>(32-bit only)</i> <i>There are known handle leak issues with earlier 9.0X versions of Ghostscript.</i>

<p>Converter Name: PEERNET Image Converter</p> <ul style="list-style-type: none"> • 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) 	<p>Built-in, no additional applications required.</p>
<p>Converter Name: Internet Explorer</p> <ul style="list-style-type: none"> • HTML Files (*.htm, *.html) • Secure HTML (*.shtm, *.shtml) • Web Archive (*.mht) 	<p>Internet Explorer 8.0 - 11.0</p>
<p>Converter Name: Microsoft Outlook</p> <ul style="list-style-type: none"> • Outlook Message Files (*.msg) • Outlook Templates (*.oft) • vCard Files (*.vcf) • vCalendar Appointment Files (*.vcs) • iCalendar Appointment Files (*.ics) 	<p>Microsoft Office 2003 <i>(*.oft and *.msg only)</i></p> <p>Microsoft Office 2007</p> <p>Microsoft Office 2010 <i>(32-bit and 64-bit)</i></p> <p>Microsoft Office 2013 <i>(32-bit and 64-bit)</i></p> <p>Microsoft Office 2016 <i>(32-bit and 64-bit)</i></p>
<p>Converter Name: Outside-In AX</p> <p>Supports over 500 common file formats; see the documentation that came with your Outside In Technology product.</p>	<p>Oracle Outside In Viewer Technology (ActiveX)</p>
<p>Converter Name: Microsoft PowerPoint</p> <ul style="list-style-type: none"> • PowerPoint Presentations (*.pptx, *.pptm, *.ppt) • PowerPoint Shows (*.ppsx, *.ppsm, *.pps) • PowerPoint Templates (*.potx, *.potm, *.pot) 	<p>Microsoft Office 2003 SP3 <i>(with Microsoft Office Compatibility Pack)</i></p> <p>Microsoft Office 2007</p> <p>Microsoft Office 2010 <i>(32-bit and 64-bit)</i></p> <p>Microsoft Office 2013 <i>(32-bit and 64-bit)</i></p> <p>Microsoft Office 2016 <i>(32-bit and 64-bit)</i></p>
<p>Converter Name: Microsoft Publisher</p> <ul style="list-style-type: none"> • Publisher Files (*.pub) 	<p>Microsoft Office 2003 SP3 <i>(with Microsoft Office Compatibility Pack)</i></p> <p>Microsoft Office 2007</p> <p>Microsoft Office 2010 <i>(32-bit and 64-bit)</i></p> <p>Microsoft Office 2013 <i>(32-bit and 64-bit)</i></p>

<p>Converter Name: Microsoft Visio</p> <ul style="list-style-type: none"> • Visio Drawings (*.vsd) 	<p>Microsoft Office 2016 (32-bit and 64-bit)</p> <p>Microsoft Visio 2003 Microsoft Visio 2007 Microsoft Visio 2010 (32-bit and 64-bit) Microsoft Visio 2013 (32-bit and 64-bit) Microsoft Visio 2016 (32-bit and 64-bit)</p>
<p>Converter Name: Microsoft Word</p> <ul style="list-style-type: none"> • Word Documents (*.docx, *.docm, *.doc) • Word Templates (*.dotx, *.dotm, *.dot) • Rich Text Documents (*.rtf) • Plain Text Files (*.txt) • Plain Text Log Files (*.log) 	<p>Microsoft Office 2003 SP3 (with Microsoft Office Compatibility Pack)</p> <p>Microsoft Office 2007</p> <p>Microsoft Office 2010 (32-bit and 64-bit)</p> <p>Microsoft Office 2013 (32-bit and 64-bit)</p> <p>Microsoft Office 2016 (32-bit and 64-bit)</p>
<p>Converter Name: Microsoft XPS</p> <ul style="list-style-type: none"> • XPS Documents (*.xps) • Open XPS Documents (*.oxps) 	<p>Uses Windows built-in XPS document support, no additional applications required.</p>
<p>Converter Name: PEERNET Passthrough</p> <ul style="list-style-type: none"> • Any file type 	<p>Built-in, passes the file through the system without converting.</p>

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.

PNDocConvQueueServiceLib COM Object

The PNDocConvQueueServiceLib COM object uses a list of name-value pairs of conversion settings to configure the output that is created. These settings are passed into the COM object directly through its `IPNDocConvQueueItem.Set` method before calling `IPNDocConvQueueItem.Convert`.

The following code sample shows the conversion settings strings for setting the resolution to 200 DPI and creating multipaged black and white TIFF files. The Resolution setting is part of the [Devmode settings](#) configuration options, while Output File Format, Append, Color reduction, and Dithering method are part of the [Save](#) configuration options.

You can find more sample output configurations by looking at the name and value pairs used in the sample [conversion profiles](#) included with Document Conversion Service.



Code Sample - C#

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("Devmode settings;Resolution", "200");
item.Set("Save;Output File Format", "TIFF Multipaged");
item.Set("Save;Color reduction", "BW");
item.Set("Save;Dithering method", "Floyd");
item.Set("TIFF File Format;BW compression", "Group4");
item.Set("TIFF File Format;Color compression", "LZW RGB");
item.Set("TIFF File Format;Indexed compression", "LZW");
item.Set("TIFF File Format;Greyscale compression", "LZW");

// convert the file
item.Convert("Microsoft Word",
            "C:\Test\Report.docx",
            "C:\Test\Out\ConvertedReport");
```

**Code Sample - VB.NET**

```
Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

' Set conversion settings
item.Set("Devmode settings;Resolution", "200")
item.Set("Save;Output File Format", "TIFF Multipaged")
item.Set("Save;Color reduction", "BW")
item.Set("Save;Dithering method", "Floyd")
item.Set("TIFF File Format;BW compression", "Group4")
item.Set("TIFF File Format;Color compression", "LZW RGB")
item.Set("TIFF File Format;Indexed compression", "LZW")
item.Set("TIFF File Format;Greyscale compression", "LZW")

' convert the file
item.Convert("Microsoft Word", _
            "C:\Test\Report.docx", _
            "C:\Test\Out\ConvertedReport")
```

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.
Endorsement Options	Endorsements are header and footer information that can be stamped onto each page of the output created by Document Conversion Service.
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.

Options	Description of Settings
PDF File Format	PDF file format settings for compression, content encoding and PDF/A-1b compliant PDF files.
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.

Creating and Customizing Profiles

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](#).

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.
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	Creates color JPEG images (one image for each page) at the dots per inch (dpi) specified. JPEG files are always serialized.

Profile Name	Profile Description
JPEG 300dpi Color JPEG 600dpi Color	
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 PDF A-1b 300dpi 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.
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.
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.

Profile Name	Profile Description
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.
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.

Profile Name	Profile Description
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.

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.

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.

Code Sample - C#

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("PageRange", "1-3");
item.Set("Devmode settings;Resolution", "300");
item.Set("Save;Output File Format", "TIFF Multipaged");
...
// convert the file
item.Convert("Microsoft Word",
             @"C:\Test\Report.docx",
             @"C:\Test\Out\ConvertedReport");
```

Code Sample - VB.NET

```
Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

' Set conversion settings
item.Set("PageRange", "1-3")
item.Set("Devmode settings;Resolution", "300")
item.Set("Save;Output File Format", "TIFF Multipaged")
...
' convert the file
item.Convert("Microsoft Word", _
            "C:\Test\Report.docx", _
            "C:\Test\Out\ConvertedReport")
```

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.

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. 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.

Code Sample - C#

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("Endorsements;HeaderHeightInPoints", "20");
item.Set("Endorsements;HeaderLeftFormat",
        "&KFF0000&BInternal Use&B");
item.Set("Endorsements;HeaderRightFormat",
        "Confidential\nDO NOT COPY");

item.Set("Endorsements;FooterHeightInPoints", "20");
item.Set("Endorsements;FooterCenterFormat",
        "&'Courier'&P of &N");
...
// convert the file
item.Convert("Microsoft Word",
            @"C:\Test\Report.docx",
            @"C:\Test\Out\ConvertedReport");
```



Code Sample - VB.NET

```

Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

' Set conversion settings
item.Set("Endorsements;Enable", "1")

item.Set("Endorsements;HeaderHeightInPoints", "20")
item.Set("Endorsements;HeaderLeftFormat", _
"&KFF0000&BInternal Use&B")
item.Set("Endorsements;HeaderRightFormat", _
"Confidential\nDO NOT COPY")

item.Set("Endorsements;FooterHeightInPoints", "20")
item.Set("Endorsements;FooterCenterFormat", _
"&'Courier'&P of &N")
...
' convert the file
item.Convert("Microsoft Word", _
"C:\Test\Report.docx", _
"C:\Test\Out\ConvertedReport")
    
```

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.
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.

Conversion Settings - Endorsements Header and Footer Options	
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.

Conversion Settings - Endorsements Header and Footer Options

Name: Endorsements;FooterLeftFormat

Values: The text, with [Endorsement Formatting Codes](#) as needed, to put in the left section of the footer.

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: <pre><add Name="Endorsements;HeaderLeftFormat" Value="Page &amp;P"/></pre> </p> <p>String Example: <pre>item.Set("Endorsements;HeaderLeftFormat", "Page &P")</pre> </p>
<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: <pre><add Name="Endorsements;HeaderLeftFormat" Value="Page &amp;P of &amp;N"/></pre> </p> <p>String Example: <pre>item.Set("Endorsements;HeaderLeftFormat", "Page &P of &N")</pre> </p>
<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: <pre><add Name="Endorsements;HeaderLeftFormat" Value="&amp;BInternal Use&amp;B - Confidential"/></pre> </p> <p>String Example: <pre>item.Set("Endorsements;HeaderLeftFormat", "&BInternal Use Only&B - Confidential")</pre> </p>
<code>&amp;l</code>	<code>&l</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: <pre><add Name="Endorsements;HeaderLeftFormat" Value="&amp;lIDo Not Copy&amp;l - Confidential"/></pre> </p> <p>String Example: <pre>item.Set("Endorsements;HeaderLeftFormat",</pre> </p>

Header and Footer Formatting Codes		
XML Code	String Code	Description
		<code>"&IDo Not Copy&I - Confidential")</code>
<code>&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: <code><add Name="Endorsements;HeaderLeftFormat" Value="&U Do Not Copy&U - Confidential"/></code></p> <p>String Example: <code>item.Set("Endorsements;HeaderLeftFormat", "&U Do Not Copy&U - Confidential")</code></p>
<code>&S</code>	<code>&S</code>	<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: <code><add Name="Endorsements;HeaderLeftFormat" Value="&S Internal Use&S - Confidential"/></code></p> <p>String Example: <code>item.Set("Endorsements;HeaderLeftFormat", "&S Internal Use Only&S - Confidential")</code></p>
<code>&X</code>	<code>&X</code>	<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: <code><add Name="Endorsements;HeaderLeftFormat" Value="This is &Xsuperscript text&X - Confidential"/></code></p> <p>String Example: <code>item.Set("Endorsements;HeaderLeftFormat", "This is &Xsuperscript text&X - Confidential")</code></p>
<code>&Y</code>	<code>&Y</code>	<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>

Header and Footer Formatting Codes		
XML Code	String Code	Description
		<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="&n14This is Arial 14 point."/></pre> <p>String Example:</p> <pre>item.Set("Endorsements;HeaderLeftFormat", "&n14This is Arial 14 point.")</pre>
&K000000	&K000000	<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 &KFF0000Red, this is &K00FF00Green."/></pre>

Header and Footer Formatting Codes		
XML Code	String Code	Description
		<p>String Example:</p> <pre>item.Set("Endorsements;HeaderLeftFormat", "This is &KFF0000Red, this is &K00FF00Green.")</pre>
&&	&&	<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>

	\n	<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\nLine 2.")</pre>

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.

Code Sample - C#

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("Microsoft.Word.Document.PrintOut.Item", "DocumentAndMarkup");
item.Set("Microsoft.Word.PageSetup.TwoPagesOnOne", "True");
item.Set("Microsoft.Word.ReplaceFieldDateWith", "<AUTODATE>");
item.Set("Devmode settings;Resolution", "300");
item.Set("Save;Output File Format", "TIFF Multipaged");
...
// convert the file
item.Convert("Microsoft Word",
             @"C:\Test\Report.docx",
             @"C:\Test\Out\ConvertedReport");
```

Code Sample - VB.NET

```
Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

' Set conversion settings
item.Set("Microsoft.Word.Document.PrintOut.Item", "DocumentAndMarkup")
item.Set("Microsoft.Word.PageSetup.TwoPagesOnOne", "True")
item.Set("Microsoft.Word.ReplaceFieldDateWith", "<AUTODATE>")
item.Set("Devmode settings;Resolution", "300")
item.Set("Save;Output File Format", "TIFF Multipaged")
...
' convert the file
item.Convert("Microsoft Word", _
            "C:\Test\Report.docx", _
            "C:\Test\Out\ConvertedReport")
```

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.
Name:	Microsoft.Word.ActiveWindow.View.RevisionsView 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.ShowComments Pass <i>True</i> to display any comments in the document. Must be used with <i>Microsoft.Word.ActiveWindow.View.MarkupMode</i> to display the comments as balloons or inline, and <i>Microsoft.Word.Document.PrintOut.Item</i> set to print document markup.
Values:	String value " True " or " False ".

Conversion Settings - Word Printing Options	
Name:	Microsoft.Word.ActiveWindow.View.ShowFormatChanges
	Pass <i>True</i> to display any formatting changes made to a document with Track Changes enabled. Must be used with <i>Microsoft.Word.ActiveWindow.View.MarkupMode</i> to display the comments as balloons or inline, and <i>Microsoft.Word.Document.PrintOut.Item</i> set to print document markup.
Values:	String value "True" or "False".
Name:	Microsoft.Word.ActiveWindow.View.ShowHiddenText
	Pass <i>True</i> to display any text that was formatted as hidden.
Values:	String value "True" or "False".
Name:	Microsoft.Word.ActiveWindow.View.ShowHighlight
	Pass <i>True</i> to have highlighted text displayed with the highlighted background.
Values:	String value "True" or "False".
Name:	Microsoft.Word.ActiveWindow.View.ShowInkAnnotations
	Pass <i>True</i> to to show handwritten ink annotations in the document. Must be used with <i>Microsoft.Word.Document.PrintOut.Item</i> set to print document markup.
Values:	String value "True" or "False".
Name:	Microsoft.Word.ActiveWindow.View.ShowInsertionsAndDeletions
	Pass <i>True</i> to display any insertions and deletions made to a document with Track Changes enabled. Must be used with <i>Microsoft.Word.ActiveWindow.View.MarkupMode</i> set to display the changes as balloons or inline, and <i>Microsoft.Word.Document.PrintOut.Item</i> set to print document markup.
Values:	String value "True" or "False".

Conversion Settings - Word Printing Options	
Name:	Microsoft.Word.ActiveWindow.View.ShowMarkupAreaHighlight
	Pass <i>True</i> to have the markup area that shows revision and comment balloons displayed shaded. Applies only when <i>Microsoft.Word.ActiveWindow.View.MarkupMode</i> is set to display markup as balloons, and <i>Microsoft.Word.Document.PrintOut.Item</i> is set to print document markup.
Values:	String value "True" or "False".
Name:	Microsoft.Word.Options.AllowA4LetterResizing
	Pass <i>True</i> 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.
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 <i>True</i> 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

Conversion Settings - Word Page Setup Printing Options

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.

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 *True* to have the inside and outside margins of facing pages to be the same width.

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

Conversion Settings - Word Page Setup Printing Options

Name: **Microsoft.Word.PageSetup.OddAndEvenPagesHeaderFooter**
Pass *True* 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

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.

Conversion Settings - Word Page Setup Printing Options

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.)

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

Set the size of the right margin in points.

Values:

String value of the desired margin width.

Conversion Settings - Word Page Setup Printing Options	
Name:	Microsoft.Word.PageSetup.SuppressEndnotes Pass <i>True</i> 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 <i>True</i> 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.



Code Sample - C#

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("Microsoft.Excel.PrintOut", "PrintOutChartsThenWorkbook");
item.Set("Microsoft.Excel.PageSetup.PrintGridlines", "True");

// Replace header/footer date fields with <AUTODATE> string
item.Set("Microsoft.Excel.ReplaceFieldDateWith", "<AUTODATE>");
item.Set("Microsoft.Excel.PageSetup.LeftHeader", "Sheet: &A");

item.Set("Devmode settings;Resolution", "300");
item.Set("Save;Output File Format", "TIFF Multipaged");
...
// convert the file
item.Convert("Microsoft Excel",
            @"C:\Test\Report.xlsx",
            @"C:\Test\Out\ConvertedReport");
```



Code Sample - VB.NET

```
Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

' Set conversion settings
item.Set("Microsoft.Excel.PrintOut", "PrintOutChartsThenWorkbook")
item.Set("Microsoft.Excel.PageSetup.PrintGridlines", "True")

' Replace header/footer date fields with <AUTODATE> string
item.Set("Microsoft.Excel.ReplaceFieldDateWith", "<AUTODATE>")
item.Set("Microsoft.Excel.PageSetup.LeftHeader", "Sheet: &A");

item.Set("Devmode settings;Resolution", "300")
item.Set("Save;Output File Format", "TIFF Multipaged")
...
' convert the file
item.Convert("Microsoft Excel", _
            "C:\Test\Report.xlsx", _
            "C:\Test\Out\ConvertedReport")
```

Conversion Settings - Excel General Formatting & Printing Options	
Name:	<p>Microsoft.Excel.PrintOut</p> <p>Choose what part of the Excel spreadsheet to print. The settings</p> <p>For <i>PrintOutChartsOnly</i>, <i>PrintOutChartsThenWorkbook</i> and <i>PrintOutWorkbookThenCharts</i>, the option <i>Microsoft.Excel.PrintOut.PrintEmbeddedChartsFirst</i> controls if embedded charts are printed before or after any chart tabs in the spreadsheet.</p>
Values:	<p>PrintOutWorkbookOnly - prints the entire workbook just as Excel does.</p> <p>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.</p> <p>PrintOutSelectedSheetsOnly - prints only the selected sheets in the workbook. Multiple sheets can be selected using the Ctrl+Left Click with the mouse.</p> <p>PrintOutSheetsWithPrintAreasOnly - prints only sheets that have a print area set.</p> <p>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.</p> <p>For the three options above, embedded charts can be before or after other charts, as specified by the <i>Microsoft.Excel.PrintOut.PrintEmbeddedChartsFirst</i> setting.</p>
Name:	<p>Microsoft.Excel.PrintHiddenWorksheets</p> <p>Choose whether to print hidden worksheets or not.</p>
Values:	<p>False - do not print hidden worksheets. True - print hidden worksheets.</p>
Name:	<p>Microsoft.Excel.PrintOut.PrintEmbeddedChartsFirst</p> <p>When printing embedded charts, determines if the embedded charts are printed before or after any chart tabs in the spreadsheet. Applies only when <i>Microsoft.Excel.PrintOut</i> is set to print charts.</p>
Values:	<p>False - print embedded charts after all other charts. True - print embedded charts first.</p>

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.Exce.PrintLastNSheets* and *Microsoft.Excel.PrintIfSheetNameMatchesRegex*.

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

Conversion Settings - Excel General Formatting & Printing Options

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.

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.

Conversion Settings - Excel General Formatting & Printing Options

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".

Conversion Settings - Excel General Formatting & Printing Options	
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.
Name:	Microsoft.Excel.AutoFitColumns
	If set to <i>True</i> 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 <i>Microsoft.Excel.AutoFitColumns</i> 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".

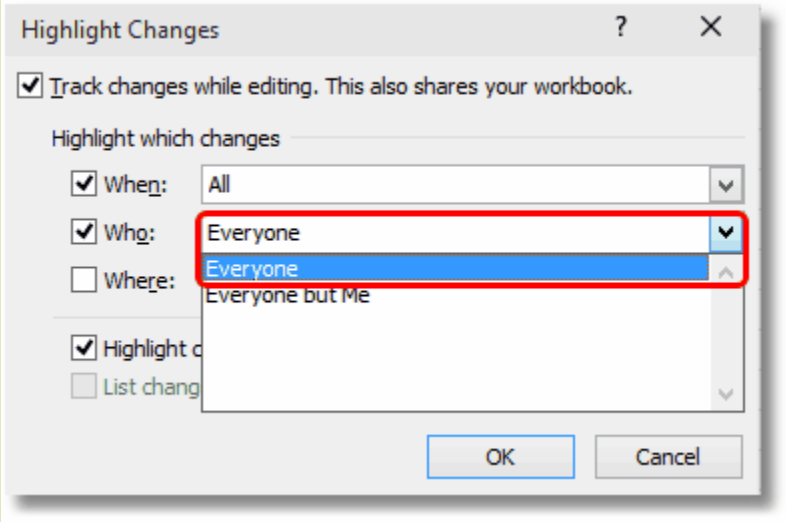
Conversion Settings - Excel General Formatting & Printing Options	
Name:	Microsoft.Excel.ClearFormatsOnEmptyRowsOnTop Clears the formatting of any empty rows (cells with no data) at the top of the spreadsheet so that only rows with data in them are printed.
Values:	String value "True" or "False".
Name:	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".
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".

Conversion Settings - Excel General Formatting & Printing Options	
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 <i>Microsoft.Excel.ClearTableStyleAndFormatting</i> .
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".
Name:	Microsoft.Excel.TrackChanges.HighlightChangesOnScreen
	If <i>Track Changes</i> 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 <i>Track Changes</i> has been enabled for the workbook, setting this to <i>True</i> will create a new temporary, protected spreadsheet that lists all of changes made to the workbook. If not using the English version of Excel, <i>Microsoft.Excel.TrackChanges.ExcelTrackChangesWhoParameter</i> will also need to be set.
Values:	String value "True" or "False".

Conversion Settings - Excel General Formatting & Printing Options

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**".

Name: **Microsoft.Excel.PageSetup.BlackAndWhite**

Print the Excel document in black and white.

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

Conversion Settings - Excel Page Setup Printing Options

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 *True* a different header or footer is used for the first page of the worksheet (*applies to Office 2007 or higher*).

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

Name: **Microsoft.Excel.PageSetup.Draft**

Prints the worksheet without graphics when set to *True*.

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 <i>True</i> .
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 <i>True</i> .
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.

Conversion Settings - Excel Page Setup Printing Options	
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.
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.)

Conversion Settings - Excel Page Setup Printing Options	
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.
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 <i>True</i> then grid lines will be printed on each spreadsheet.
Values:	String value "True" or "False".
Name:	Microsoft.Excel.PageSetup.PrintHeadings
	If set to <i>True</i> then column and row headings will be printed on each spreadsheet.
Values:	String value "True" or "False".
Name:	Microsoft.Excel.PageSetup.PrintNotes
	Set to <i>True</i> 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 <i>DevMode settings;Resolution</i> setting in the Devmode settings section.

Conversion Settings - Excel Page Setup Printing Options	
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.
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 <i>True</i> then the header and footer will be scaled with the document when the size of the document changes.
Values:	String value "True" or "False".

Conversion Settings - Excel Page Setup Printing Options

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.

Conversion Settings - Excel Field Replacement	
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.
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.

Conversion Settings - Document Protection

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.

Header and Footer Formatting Codes

&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.

Code Sample - C#

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("Microsoft.PowerPoint.PageSetup.FirstSlideNumber", "2");
item.Set("Microsoft.PowerPoint.PageSetup.NotesOrientation",
        "OrientationVertical");
item.Set("Microsoft.PowerPoint.PrintOptions.FitToPage",
        "True");
item.Set("Devmode settings;Resolution", "300");
item.Set("Save;Output File Format", "TIFF Multipaged");
...
// convert the file
item.Convert("Microsoft PowerPoint", _
            @"C:\Test\Report.pptx", _
            @"C:\Test\Out\ConvertedPresentation");
```

Code Sample - VB.NET

```
Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

' Set conversion settings
item.Set("Microsoft.PowerPoint.PageSetup.FirstSlideNumber", "2")
item.Set("Microsoft.PowerPoint.PageSetup.NotesOrientation",
        "OrientationVertical")
item.Set("Microsoft.PowerPoint.PrintOptions.FitToPage",
        "True")
item.Set("Devmode settings;Resolution", "300")
item.Set("Save;Output File Format", "TIFF Multipaged")
...
' convert the file
item.Convert("Microsoft PowerPoint", _
            "C:\Test\Report.pptx", _
            "C:\Test\Out\ConvertedPresentation")
```

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".

Conversion Settings - PowerPoint Page Setup	
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.
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

Conversion Settings - PowerPoint Page Setup

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".


Conversion Settings - PowerPoint Print Options**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.

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.


 **Code Sample - C#**

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("Adobe.PDF.PrintOptions.CommentsAndForms", "DocumentsAndMarkups");

item.Set("Devmode settings;Resolution", "300");
item.Set("Save;Output File Format", "TIFF Multipaged");
...
// convert the file
item.Convert("Adobe Acrobat Reader", _
            @"C:\Test\Report.pdf", _
            @"C:\Test\Out\ConvertedPDF");
```

 **Code Sample - VB.NET**

```
Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

' Set conversion settings
item.Set("Adobe.PDF.PrintOptions.CommentsAndForms", "DocumentsAndMarkups")
item.Set("Devmode settings;Resolution", "300")
item.Set("Save;Output File Format", "TIFF Multipaged")
...
' convert the file
item.Convert("Adobe Acrobat Reader", _
            "C:\Test\Report.pdf", _
            "C:\Test\Out\ConvertedPDF")
```

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:	<p>DocumentsAndMarkups - prints the document with any markup and stamps visible.</p> <p>DocumentsAndStamps - prints the document with only stamp annotations visible. Markup is not shown</p> <p>Documents - prints only the document. Markup and stamps are not printed.</p>

Conversion Settings - Adobe Reader Print Options

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 <i>false</i> . 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 <i>Adobe.PDF.PrintOptions.ChoosePaperSourceByPDFPageSize</i> is set to "False". This option is disabled (set to "False") by default.
Values:	String value "True" or "False".
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".

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".

Internet Explorer Options

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.

The default Internet Explorer options are to print no headers or footer information, use margins of 0.75", to print all background color and images and to shrink the page to fit. See [Adding Headers, Footers and Fonts to HTML Conversion](#) for instruction on customizing the Internet Explorer converter settings.

There are also application level Internet Explorer settings to control image scaling and browser emulation; see [Application Level Configuration Settings](#) to change these options.

Code Sample - C#

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("Microsoft.InternetExplorer.PageSetup.Footer",
        "&b&u&b");
item.Set("Microsoft.InternetExplorer.PageSetup.MarginBottom", "0.50");
item.Set("Microsoft.InternetExplorer.PageSetup.MarginLeft", "0.50");
item.Set("Microsoft.InternetExplorer.PageSetup.MarginRight", "0.50");
item.Set("Microsoft.InternetExplorer.PageSetup.MarginTop", "0.50");
...
// convert the file
item.Convert("Internet Explorer",
            @"C:\Test\ArchiveReport.mht",
            @"C:\Test\Out\ConvertedReport");
```

Code Sample - VB.NET

```
Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

' Set conversion settings
item.Set("Microsoft.InternetExplorer.PageSetup.Footer", "&b&u&b")
item.Set("Microsoft.InternetExplorer.PageSetup.MarginBottom", "0.50")
item.Set("Microsoft.InternetExplorer.PageSetup.MarginLeft", "0.50")
item.Set("Microsoft.InternetExplorer.PageSetup.MarginRight", "0.50")
item.Set("Microsoft.InternetExplorer.PageSetup.MarginTop", "0.50")

...
' convert the file
item.Convert("Internet Explorer", _
            "C:\Test\ArchiveReport.mht", _
            "C:\Test\Out\ConvertedReport")
```

Conversion Settings - Page Setup	
Name:	Microsoft.InternetExplorer.PageSetup.Header
	The format of the header to print on each page. By default, no page header is printed.
Values:	If you do want a header when converting HTML files, follow the instructions here .
Name:	Microsoft.InternetExplorer.PageSetup.Footer
	The format of the footer to print on each page. By default, no page footer is printed.
Values:	If you do want a footer when converting HTML files, follow the instructions here .
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, <i>font-style: italic</i> ; and <i>font-weight: bold</i> 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) ; <i>font-style: italic</i> ; <i>font-weight: bold</i> ;
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.

Conversion Settings - Page Setup**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".**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](#).

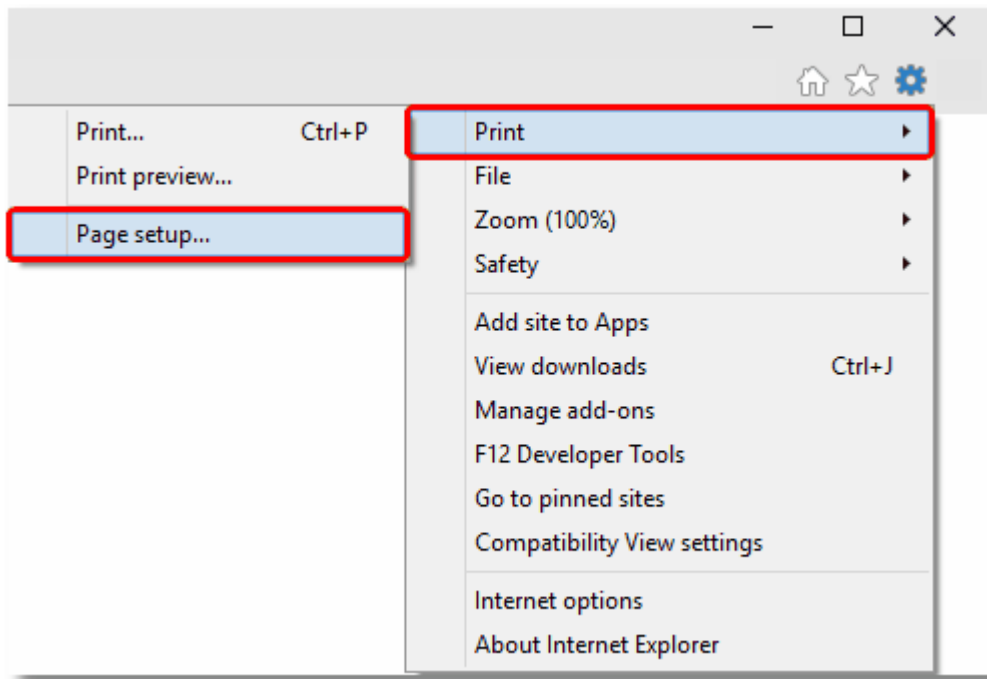
```
<AppFactory Name="Internet Explorer"
  Type="PEERNET.PNDocConv.Applications.PNInternetExplorerApplicatio
  Assembly="PNInternetExplorerApplicationFactory">
  <Settings>
    ...
    <add Name="ConverterPlugIn.PNIExplorer.ShrinkToFitScaleMin" Value="30" />
  </Settings>
</AppFactory>
```

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

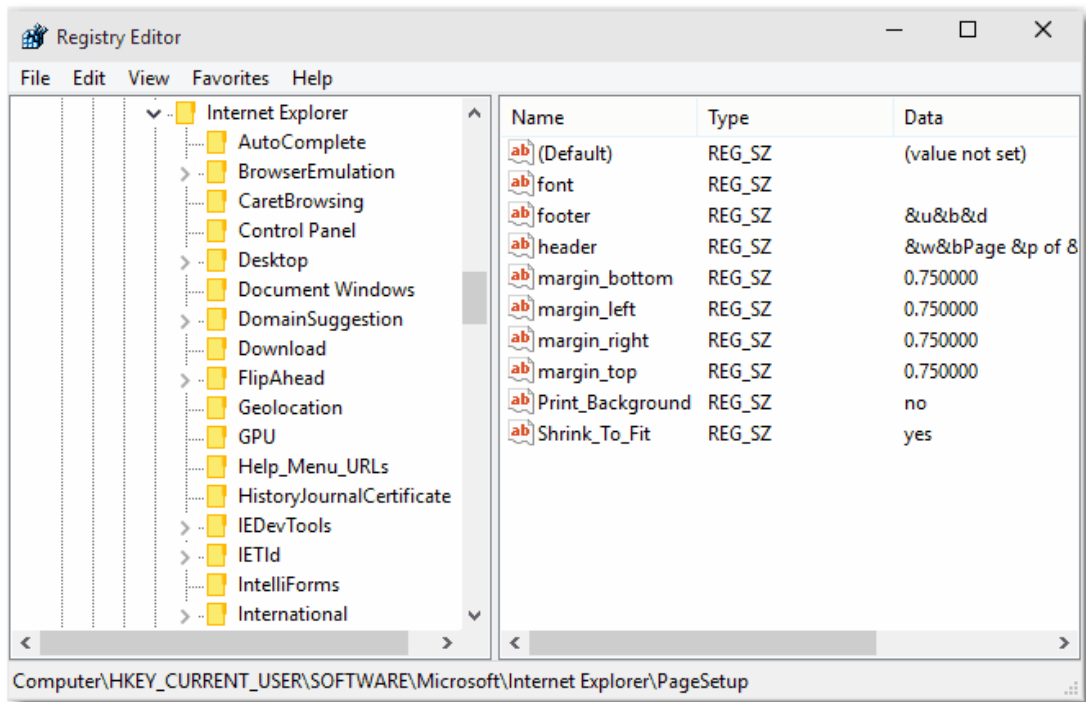
Adding Headers, Footers and Fonts to HTML Conversion

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.

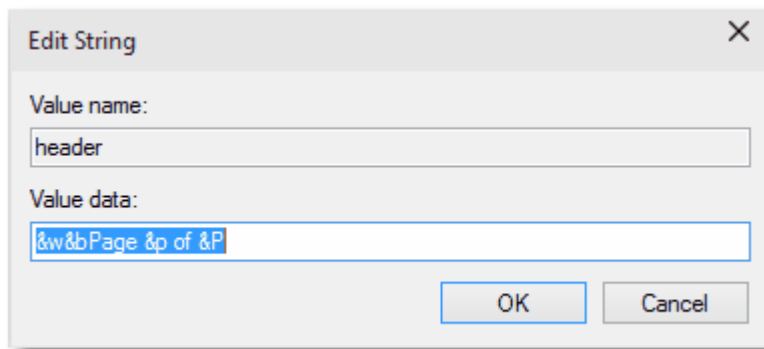
1. Open Internet Explorer to any web page or html file.
2. In the upper right corner, click the Tools icon (it looks like a blue gear), then select Print - Page Setup.
 - a. Alternatively you can press the F10 key to show the application menu and then select File - Page Setup.



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.



- 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.PNIExplorer.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.PNIExplorer.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.PNIExplorer.ShrinkToFitScaleMin" Value="30"/>

        <!-- Values: Empty string, IE7, IE8, IE8FORCE, IE9, IE9FORCE, IE10, IE10FORCE, IE11 -->
        <add Name="ConverterPlugIn.PNIExplorer.BrowserEmulation" Value="" />


      </Settings>
    </AppFactory>

    ...

  </Factories>
</Settings>
<!-- Global factory settings -->
<add Name="MaxInstances" Value="auto"/>
<add Name="RecycleThreshold" Value="0"/>
</Settings>
</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.


 **Code Sample - C#**

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("ConverterPlugIn.PNGghostscriptConverter.TextAntiAlias", "4");
item.Set("ConverterPlugIn.PNGghostscriptConverter.Graphics", "4");

item.Set("ConverterPlugIn.PNGghostscriptConverter.FontPath",
        @"C:\psfonts;c:\Windows\Fonts;C:\MyFonts");
...
// convert the file
item.Convert("Ghostscript",
            @"C:\Test\ArchiveReport.ps",
            @"C:\Test\Out\ConvertedReport");
```

 **Code Sample - VB.NET**

```
Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

' Set conversion settings
item.Set("ConverterPlugIn.PNGghostscriptConverter.TextAntiAlias", "4")
item.Set("ConverterPlugIn.PNGghostscriptConverter.Graphics", "4")

item.Set("ConverterPlugIn.PNGghostscriptConverter.FontPath", _
        "C:\psfonts;c:\Windows\Fonts;C:\MyFonts")

...
' convert the file
item.Convert("Ghostscript", _
            "C:\Test\ArchiveReport.ps", _
            "C:\Test\Out\ConvertedReport")
```

Conversion Settings - Page Setup	
Name:	ConverterPlugIn.PNGghostscriptConverter.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 *Fonts* 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.

 **Code Sample - C#**

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;


// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("ConverterPlugIn.PNImageConverter.ImageToolkitOrder", "LEAD;WIC");
item.Set("ConverterPlugIn.PNImageConverter.LEADScalingMode", "BICUBIC");
item.Set("ConverterPlugIn.PNImageConverter.WICScalingMode", "BICUBIC");

// Background color for transparent images, white is default
item.Set("ConverterPlugIn.PNImageConverter.AlphaBackgroundColorRGB", "255,255,255");

// Output file options
item.Set("Devmode settings;Resolution", "300");
item.Set("Save;Output File Format", "TIFF Multipaged");

...
// convert the file
item.Convert("PEERNET Image Converted",
            @"C:\Test\screenshot.png",
            @"C:\Test\Out\ConvertedImage");
```

 **Code Sample - VB.NET**

```
Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

' Set conversion settings
item.Set("ConverterPlugIn.PNImageConverter.ImageToolkitOrder", "LEAD;WIC")
item.Set("ConverterPlugIn.PNImageConverter.LEADScalingMode", "BICUBIC")
item.Set("ConverterPlugIn.PNImageConverter.WICScalingMode", "BICUBIC")

' Background color for transparent images, white is default
item.Set("ConverterPlugIn.PNImageConverter.AlphaBackgroundColorRGB", "255,255,255")

item.Set("Devmode settings;Resolution", "300")
item.Set("Save;Output File Format", "TIFF Multipaged")

...
' convert the file
item.Convert("PEERNET Image Converted", _
            "C:\Test\screenshot.png", _
            "C:\Test\Out\ConvertedImage")
```


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 (**Windows Imaging Component**) 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.

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.

Conversion Settings - Toolkits and Scaling Modes	
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 <i>Devmode settings;Resolution</i> 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 <i>Devmode settings;Resolution</i> setting.
Name:	ConverterPlugIn.PNImageConverter.ResampleImageToMaxWidthOrHeightInPixels
	Dynamically sample the output image to a specific maximum width or height, whichever criteria is met first. The desired dimension is specified in <i>pixels</i> . 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.

Image Format	LEAD	WIC
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.

Code Sample - C#

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("Oracle.OutsideInAX.BMPPrintBorder", "0");
item.Set("Oracle.OutsideInAX.IntlFlags", "1");
item.Set("Oracle.OutsideInAX.PrintMarginTop", "0.50");
item.Set("Oracle.OutsideInAX.PrintMarginBottom", "0.50");

item.Set("Devmode settings;Resolution", "300");
item.Set("Save;Output File Format", "TIFF Multipaged");
...
// convert the file
item.Convert("Microsoft Outside-In AX",
            "C:\Test\Report.wpd",
            "C:\Test\Out\ConvertedReport");
```

Code Sample - VB.NET

```
Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

' Set conversion settings
item.Set("Oracle.OutsideInAX.BMPPrintBorder", "0")
item.Set("Oracle.OutsideInAX.IntlFlags", "1")
item.Set("Oracle.OutsideInAX.PrintMarginTop", "0.50")
item.Set("Oracle.OutsideInAX.PrintMarginBottom", "0.50")

item.Set("Devmode settings;Resolution", "300")
item.Set("Save;Output File Format", "TIFF Multipaged")
...
' convert the file
item.Convert("Microsoft Word", _
            "C:\Test\Report.wpd", _
            "C:\Test\Out\ConvertedReport")
```

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.
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.

Conversion Settings - OutsideIn AX Printing

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.

Code Sample - C#

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("Devmode settings;Resolution", "300");
item.Set("Save;Output File Format", "TIFF Serialized");
item.Set("Save;Color reduction", "BW");
item.Set("Save;
...
// convert the file
item.Convert("Microsoft Word",
             @"C:\Test\Report.docx",
             @"C:\Test\Out\ConvertedReport");
```

Code Sample - VB.NET

```
Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

' Set conversion settings
item.Set("Devmode settings;Resolution", "300")
item.Set("Save;Output File Format", "TIFF Serialized")
item.Set("Save;Prompt", "0")
item.Set("Save;Overwrite", "1")
item.Set("Save;Color reduction", "BW")
...
' convert the file
item.Convert("Microsoft Word", _
            "C:\Test\Report.docx", _
            "C:\Test\Out\ConvertedReport")
```

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.

Conversion Settings - Save	
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.
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 <i>Document.doc</i> created as TIFF would become <i>Document.doc.tif</i> ; when set to remove the extension, the resulting filename would be <i>Document.tif</i> .
Values:	0 - Leave original filename extension in new filename 1 - Remove original filename extension before creating new filename.

Conversion Settings - Save	
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
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 <i>SplitFileWhenFileSizeExceedsThresholdEnabled</i> to split by page count and file size.
	File splitting only applies to the following multipaged file formats: <ul style="list-style-type: none"> • 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.

Conversion Settings - Save	
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 <i>SplitFileEveryNPagesEnabled</i> to split by file size and page count.
	File splitting only applies to the following multipaged file formats: <ul style="list-style-type: none"> • 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 <i>SplitFileSizeThresholdInBytes</i> .
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.

Code Sample - C#

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("Devmode settings;Resolution", "300");
item.Set("Devmode settings;Color", "1");
item.Set("Save;Output File Format", "TIFF Multipaged");
...
// convert the file
item.Convert("Microsoft Word",
            @"C:\Test\Report.docx",
            @"C:\Test\Out\ConvertedReport");
```

Code Sample - VB.NET

```
Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

' Set conversion settings
item.Set("Devmode settings;Resolution", "300")
item.Set("Devmode settings;Color", "1")
item.Set("Save;Output File Format", "TIFF Multipaged")
...
' convert the file
item.Convert("Microsoft Word", _
            "C:\Test\Report.docx", _
            "C:\Test\Out\ConvertedReport")
```

Conversion Settings - Devmode

Name:	Devmode settings;Orientation
	Orientation of the page when the file is converted.
Values:	Portrait Landscape

Conversion Settings - Devmode	
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 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

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.

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
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	Inclu de Jobl D	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.

Code Sample - C#

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("Devmode settings;Resolution", "300");
item.Set("Devmode settings;Color", "1");
item.Set("Save;Output File Format", "TIFF Serialized");

item.Set("Advanced File Naming;Format string S", "%s");
item.Set("Advanced File Naming;Variables S",
        "${PrintedPageNumber}");
...
// convert the file
item.Convert("Microsoft Word",
            @"C:\Test\Report.docx",
            @"C:\Test\Out\ConvertedReport");
```



Code Sample - VB.NET

```
Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem
' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()
' Set conversion settings
item.Set("Devmode settings;Resolution", "300")
item.Set("Devmode settings;Color", "1")
item.Set("Save;Output File Format", "TIFF Serialized")
item.Set("Advanced File Naming;Format string S", "%s")
item.Set("Advanced File Naming;Variables S",
        "$(PrintedPageNumber)")
...
' convert the file
item.Convert("Microsoft Word", _
            "C:\Test\Report.docx", _
            "C:\Test\Out\ConvertedReport")
```

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.

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.

Conversion Settings - Advanced File Naming	
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 <i>Advanced File Naming;Variables SJ</i> 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 <i>Advanced File Naming;Format string SJ</i> 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 <i>Advanced File Naming;Variables M</i> 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
Name:	Advanced File Naming;Variables M
	Comma-delimited list of variables that correspond to the placeholders in the format string supplied in <i>Advanced File Naming;Format string M</i> above.
Values:	See list of variables below.

Conversion Settings - Advanced File Naming	
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 <i>Advanced File Naming;Variables MJ</i> 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 <i>Advanced File Naming;Format string MJ</i> 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.

Variable	Type and Format String Place Holder	Description
\$(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.
\$(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)

Profile	Format String	Variables and Resulting File Names for TIFF Creation
		\$(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.


 **Code Sample - C#**

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("Devmode settings;Color", "1");
item.Set("Image Options;Fax", "1");
item.Set("Image Options;Fax Profile", "0");
item.Set("Image Options;Fax Resolution", "3");
item.Set("Save;Output File Format", "TIFF Serialized");

...
// convert the file
item.Convert("Microsoft Word",
             @"C:\Test\Report.docx",
             @"C:\Test\Out\ConvertedReport");
```

 **Code Sample - VB.NET**

```
Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

' Set conversion settings
item.Set("Devmode settings;Color", "1")
item.Set("Image Options;Fax", "1")
item.Set("Image Options;Fax Profile", "0")
item.Set("Image Options;Fax Resolution", "3")
item.Set("Save;Output File Format", "TIFF Serialized")

...
' convert the file
item.Convert("Microsoft Word", _
            "C:\Test\Report.docx", _
            "C:\Test\Out\ConvertedReport")
```

Conversion Settings - Image Options	
Name:	Image Options;Fax
Values:	<p>0 - Do not create fax format file.</p> <p>1 - Create an image where its width is limited to fax resolution as determined by Fax Profile and Fax Resolution settings</p>


Conversion Settings - Image Options	
Name:	Image Options;Fax Profile
Values:	<ul style="list-style-type: none"> 0 - Profile F, standard monochrome 1 - Profile S, simplified monochrome 2 - Profile C, color fax
Name:	Image Options;Fax Resolution
Values:	<ul style="list-style-type: none"> 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)
Name:	Image Options;Fax Use Printer Resolution
Values:	<ul style="list-style-type: none"> 0 - Do not use printer resolution 1 - Use printer resolution
Name:	Image Options;Fax Paper Width
Values:	<ul style="list-style-type: none"> 0 - Letter 1 - Legal 2 - A4 (ISO) 3 - B4 (ISO) 4 - A3 (ISO) 5 - Auto
Name:	Image Options;Fax Paper Height
Values:	<ul style="list-style-type: none"> 0 - Variable height 1 - Fixed height

Conversion Settings - Image Options	
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
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

Conversion Settings - Image Options	
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
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.

 **Code Sample - C#**

```

PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("Devmode settings;Color", "1");
item.Set("Save;Output File Format", "TIFF Serialized");
item.Set("Save;Color reduction", "Optimal");
item.Set("TIFF File Format;BW compression", "Group3-2D");
item.Set("TIFF File Format;Color compression", "LZW");
...
// convert the file
item.Convert("Microsoft Word",
             @"C:\Test\Report.docx",
             @"C:\Test\Out\ConvertedReport");
    
```

 **Code Sample - VB.NET**

```

Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()


' Set conversion settings
item.Set("Devmode settings;Color", "1")
item.Set("Save;Output File Format", "TIFF Serialized")
item.Set("Save;Color reduction", "Optimal")
item.Set("TIFF File Format;BW compression", "Group3-2D")
item.Set("TIFF File Format;Color compression", "LZW")
...
' convert the file
item.Convert("Microsoft Word", _
            "C:\Test\Report.docx", _
            "C:\Test\Out\ConvertedReport")
    
```

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

Conversion Settings - TIFF File Format	
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
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.


 **Code Sample - C#**

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("Devmode settings;Color", "1");
item.Set("Save;Output File Format", "PDF Multipaged");
item.Set("Save;Prompt", "0");
item.Set("Save;Overwrite", "1");
item.Set("PDF File Format;PDF Standard", "PDF/A-1b");
item.Set("PDF File Format;Use compression", "1");

...
// convert the file
item.Convert("Microsoft Word",
            @"C:\Test\Report.docx",
            @"C:\Test\Out\ConvertedReport");
```

 **Code Sample - VB.NET**

```
Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

' Set conversion settings
item.Set("Devmode settings;Color", "1")
item.Set("Save;Output File Format", "PDF Multipaged")
item.Set("Save;Prompt", "0")
item.Set("Save;Overwrite", "1")
item.Set("PDF File Format;PDF Standard", "PDF/A-1b")
item.Set("PDF File Format;Use compression", "1")

...
' convert the file
item.Convert("Microsoft Word", _
            "C:\Test\Report.docx", _
            "C:\Test\Out\ConvertedReport")
```

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

Conversion Settings - PDF File Format	
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
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

Conversion Settings - PDF File Format	
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.

Code Sample - C#

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("Devmode settings;Color", "1");
item.Set("Save;Output File Format", "PDF Multipaged");
item.Set("Save;Prompt", "0");
item.Set("Save;Overwrite", "1");
item.Set("PDF File Format;PDF Standard", "None");
item.Set("PDF File Format;Use compression", "1");
item.Set("PDF Security;Use Security", "1");
item.Set("PDF Security;Encrypt Level", "1");

...
// convert the file
item.Convert("Microsoft Word",
            @"C:\Test\Report.docx",
            @"C:\Test\Out\ConvertedReport");
```

Code Sample - VB.NET

```
Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

' Set conversion settings
item.Set("Devmode settings;Color", "1")
item.Set("Save;Output File Format", "PDF Multipaged")
item.Set("Save;Prompt", "0")
item.Set("Save;Overwrite", "1")
item.Set("PDF File Format;PDF Standard", "None")
item.Set("PDF File Format;Use compression", "1")
item.Set("PDF Security;Use Security", "1")
item.Set("PDF Security;Encrypt Level", "1")

...
' convert the file
item.Convert("Microsoft Word", _
            "C:\Test\Report.docx", _
            "C:\Test\Out\ConvertedReport")
```

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
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.

Conversion Settings - PDF Security	
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.

Code Sample - C#

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("Devmode settings;Resolution", "300");
item.Set("Save;Output File Format", "JPEG");
item.Set("Save;Prompt", "0");
item.Set("Save;Overwrite", "1");
item.Set("Save;Color reduction", "Optimal");
item.Set("JPEG File Format;Color compression", "Medium Quality");
item.Set("JPEG File Format;Greyscale compression", "High Quality");
...
// convert the file
item.Convert("Microsoft Word",
            @"C:\Test\Report.docx",
            @"C:\Test\Out\ConvertedReport");
```

Code Sample - VB.NET

```
Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

' Set conversion settings
item.Set("Devmode settings;Resolution", "300")
item.Set("Save;Output File Format", "JPEG")
item.Set("Save;Prompt", "0")
item.Set("Save;Overwrite", "1")
item.Set("Save;Color reduction", "Optimal")
item.Set("JPEG File Format;Color compression", "Medium Quality")
item.Set("JPEG File Format;Greyscale compression", "High Quality")
...
' convert the file
item.Convert("Microsoft Word", _
            "C:\Test\Report.docx", _
            "C:\Test\Out\ConvertedReport")
```

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

Conversion Settings - JPEG File Format**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.

Code Sample - C#

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("Devmode settings;Resolution", "300");
item.Set("Save;Output File Format", "TIFF Serialized");
item.Set("Save;Prompt", "0");
item.Set("Processing;Trim Threshold", "0");
item.Set("Processing;Trim left", "1");
item.Set("Processing;Trim top", "1");
item.Set("Processing;Trim bottom", "1");
item.Set("Processing;Trim right", "1");
...
// convert the file
item.Convert("Microsoft Word",
            @"C:\Test\Report.docx",
            @"C:\Test\Out\ConvertedReport");
```

Code Sample - VB.NET

```
Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

' Set conversion settings
item.Set("Devmode settings;Resolution", "300")
item.Set("Save;Output File Format", "TIFF Serialized")
item.Set("Save;Prompt", "0")
item.Set("Processing;Trim Threshold", "0")
item.Set("Processing;Trim left", "1")
item.Set("Processing;Trim top", "1")
item.Set("Processing;Trim bottom", "1")
item.Set("Processing;Trim right", "1")
...
' convert the file
item.Convert("Microsoft Word", _
            "C:\Test\Report.docx", _
            "C:\Test\Out\ConvertedReport")
```

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 <i>Trim Threshold</i> 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 <i>Trim Threshold</i> below.
Values:	0 - Do not trim top of page 1 - Trim top of page
Name:	Processing;Trim right
	Trim all areas from the right side of the page, based on the <i>Trim Threshold</i> 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 <i>Trim Threshold</i> below.
Values:	0 - Do not trim bottom of page 1 - Trim bottom of page

Conversion Settings - Processing	
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 <i>crop region</i> .
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;Crop top
	Applies when Crop Option is set to 0 for <i>crop region</i> .
Values:	Same as <i>Processing;Crop left</i> above

Conversion Settings - Processing	
Name:	Processing;Crop width Applies when Crop Option is set to 0 for <i>crop region</i> .
Values:	Same as <i>Processing;Crop left above</i> .
Name:	Processing;Crop height Applies when Crop Option is set to 0 for <i>crop region</i> .
Values:	Same as <i>Processing;Crop left above</i>
Name:	Processing;Crop margin left Applies when Crop Option is set to 1 for <i>crop margins</i> .
Values:	Same as <i>Processing;Crop left above</i>
Name:	Processing;Crop margin top Applies when Crop Option is set to 1 for <i>crop margins</i>
Values:	Same as <i>Processing;Crop left above</i>
Name:	Processing;Crop margin right Applies when Crop Option is set to 1 for <i>crop margins</i>
Values:	Same as <i>Processing;Crop left above</i>
Name:	Processing;Crop margin bottom Applies when Crop Option is set to 1 for <i>crop margins</i>
Values:	Same as <i>Processing;Crop left above</i>
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 <i>Processing;Copy to width</i> above.
Name:	Processing;Copy to IAM Left The desired left area margin settings for the new image.
Values:	Same as <i>Processing;Copy to width</i> above
Name:	Processing;Copy to IAM Top The desired top area margin settings for the new image.
Values:	Same as <i>Processing;Copy to width</i> above
Name:	Processing;Copy to IAM Right The desired right area margin settings for the new image.
Values:	Same as <i>Processing;Copy to width</i> above
Name:	Processing;Copy to IAM Bottom The desired bottom area margin settings for the new image.
Values:	Same as <i>Processing;Copy to width</i> above

Conversion Settings - Processing	
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
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

Conversion Settings - Processing	
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
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

Conversion Settings - Processing	
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
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.

Code Sample - C#

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

// Create the conversion item
item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

// Set conversion settings
item.Set("Devmode settings;Resolution", "300");
item.Set("Save;Output File Format", "TIFF Serialized");
item.Set("Save;Prompt", "0");
item.Set("Advanced Features;Extract Text", "1");
item.Set("Advanced Features;Extract Text Layout", "Physical");
...
// convert the file
item.Convert("Microsoft Word",
             @"C:\Test\Report.docx",
             @"C:\Test\Out\ConvertedReport");
```

Code Sample - VB.NET

```
Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

' Create the conversion item
item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

' Set conversion settings
item.Set("Devmode settings;Resolution", "300")
item.Set("Save;Output File Format", "TIFF Serialized")
item.Set("Save;Prompt", "0")
item.Set("Advanced Features;Extract Text", "1")
item.Set("Advanced Features;Extract Text Layout", "Physical")
...
' convert the file
item.Convert("Microsoft Word", _
            "C:\Test\Report.docx", _
            "C:\Test\Out\ConvertedReport")
```

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
Name:	Advanced Features;Custom Paper Height
	Specify the height of the custom paper size. <i>Custom Paper Enable</i> 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












Conversion Settings - Advanced Features	
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
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

Conversion Settings - Advanced Features	
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

PNDocConvQueueServiceLib Object Reference

The PNDocConvQueueServiceLib COM interface provides developers with the ability to convert documents using the Document Conversion Service from their own application code.

Objects

Object	Description
 IPNDocConvQueueItem	This is the main object you will be working with when converting files.
 IPNDocConvQueueItemJobs	A collection of IPNDocConvQueueItemJob objects, one for every job that has been printed.
 IPNDocConvQueueItemJob	Each IPNDocConvQueueItemJob object represents a single print job sent to any one of the printer queues in the current IPNSession object.
 IPNDocConvQueueItemFiles	A collection of IPNDocConvQueueItemFile objects, one for every file created.
 IPNDocConvQueueItemFile	An IPNDocConvQueueItemFile object is created for every file created by the conversion service and represents the physical file on disk.
 IPNDocConvQueueItemPages	A collection of IPNDocConvQueueItemPage objects, one for every page of a document converted.
 IPNDocConvQueueItemPage	An IPNDocConvQueueItemPage object is created for every page of the converted document or file
 IPNDocConvQueueItemImages	A collection of IPNDocConvQueueItemImage objects, one for every page in the output file on disk.
 IPNDocConvQueueItemImage	An IPNDocConvQueueItemImage object represents a single page in the output physical file on disk.
 IPNDocConvQueueItemMessages	A collection of IPNDocConvQueueItemMessage objects. This collection can be empty.
 IPNDocConvQueueItemMessage	A IPNDocConvQueueItemMessage object wraps a string message that contains information from the conversion process.
 IPNDocConvQueueItemErrors	A collection of IPNDocConvQueueItemError objects. This collection will be empty if the conversion succeeds.
 IPNDocConvQueueItemError	A IPNDocConvQueueItemError object wraps a string message that contains error information from the conversion process if it failed.

IPNDocConvQueueItem

Description

The IPNDocConvQueueItem object is the main object for using the PNDocConvQueueServiceLib COM API to communicate with the Document Conversion Service.

The IPNDocConvQueueItem object also holds the settings used to convert the file. The settings are stored as a dictionary collection of *key-value* pairs where the key is a unique string for each value. The value can be any object although strings are mainly used.








Settings are added by calling the [Set](#) method for each setting you need to configure. You can test for settings using the [Contains](#) method, retrieve the settings with the [Get](#) methods and remove them using the [Remove](#) method.

If you need to convert many files using the same conversion settings you would create a single an IPNDocConvQueueItem object with the desired settings and reuse that same object for each file you want to convert.





If you need to specify different settings for each file you are converting you have the choice of altering the settings collection as needed for each file before converting, or creating a new IPNDocConvQueueItem with its own settings collection for each file.

Two functions, [WaitForRunningDocumentConversionService](#) and [WaitForAvailableConverterPlugIn](#) allow you to synchronize long running applications such as a watch folder service with the Document Conversion Service being up and running and ready to convert.








Methods

 Contains	Determines whether the settings collection contains the specified key.
 Convert	Converts the source file to the output file using the converter specified.
 Get	Gets the value associated with the specified key name.
 Remove	Removes the value with the associated key from the settings collection.
 Set	Adds the specified key and value to the collection.
 WaitForAvailableConverterPlugIn	Waits until the converter has been loaded by the Document Conversion Service or the time-out interval elapses.
 WaitForRunningDocumentConversionService	Waits until the Document Conversion Service is running or the time-out interval elapses.

Properties

 Errors	Read-only; A collection of IPNDocConvQueueItemError objects. This collection will be empty if the conversion succeeds.
 Files	Read-only; A collection of IPNDocConvQueueItemFile objects. There will be one object for each file created.
 Images	Read-only; A collection of IPNDocConvQueueItemImage objects. There will be one object for each page in each file created.
 Jobs	Read-only; A collection of IPNDocConvQueueItemJob objects. There will be one object for each print job created by the conversion service.
 Messages	Read-only; A collection of IPNDocConvQueueItemMessage objects. There will be one object for each information message reported by the conversion service. This collection can be empty.
 Pages	Read-only; A collection of IPNDocConvQueueItemPage objects. There will be one object for each page in each file created by the conversion service.

Events

 OnCloseFile	This event is fired each time a file we are creating is closed.
 OnEndConversion	This event is fired when the conversion process is complete for the source file.
 OnEndImage	This event is fired each time a new image for a page is completed.
 OnEndJob	This event is fired each time a single job for a file has completed printing. Some files create multiple jobs.
 OnEndPage	This event is fired each time a new page in an output file is completed
 OnGetNextOutputFileName	This event is fired each time a file name needs to be created but as it is easier to pass the desired output file name in the Convert method this event rarely needs to be handled.
 OnLogMessage	This event is fired when informational messages are logged by the conversion process.

 OnOpenFile	This event is fired each time a new output file is opened or created as part of the conversion process.
 OnReportError	This event is fired when an error occurs during the conversion process.
 OnStartConversion	This event is fired when the source file is received to be converted but before the conversion process is started
 OnStartImage	This event is fired each time a new image for a page is created.
 OnStartJob	This event is fired each time a single job for a file has started printing through the converters. Some files can create multiple jobs.
 OnStartPage	This event is fired each time a new page in an output file is to be created

Methods

Contains

Description

Tests whether a key exists in the settings collection.

Syntax

expression.Contains(*Key*)

where *expression* is an [IPNDocConvQueueItem](#) object.

Returns a Boolean **True** if the key exists in the settings collection, **False** if it does not.

Parameters

String Key

The key to locate in the settings collection.

Exceptions

Exception	Condition
ArgumentNullException	Key is null or empty

See Also:

[Get](#) [Set](#) [Remove](#) [Convert](#)

Convert

Description

Convert the input file to a new file with the specified name (no extension is needed) using the converter specified. The type of file created is specified by the conversion options set in the IPNDocConvQueueItem's settings collection before calling this method.

Syntax

```
expression.Convert(ConverterPlugInName, Input, Output)
```

where expression is an [IPNDocConvQueueItem](#) object.

Parameters

String ConverterPlugInName

A list of one or more converters to use to try and convert the file. Multiple converters can be specified as a semi-colon separated list. The first matching converter in the list is used. See [Converters](#) for a list of converter names that can be used.

String Input

The input file to convert. This needs to be a UNC-based path if the conversion is being done from a mapped drive.

String Output

The full path and name of the output file, without the file extension. The file extension is automatically added based on the conversion type.

Exceptions

Exception	Condition
COMException	<p>There The following errors will cause this exception. The exception message details the cause of the error.</p> <ul style="list-style-type: none"> • Document Conversion Service is not running or is unavailable • The converter specified is unavailable • The item failed to convert in the timeout set by Document Conversion Service

See Also:

[Contains Get Set Remove Converters](#)

Examples



Code Sample - C#

```
PNDocConvQueueServiceLib.PNDocConvQueueItem item = null;

try
{
    // Create the conversion item
    item = new PNDocConvQueueServiceLib.PNDocConvQueueItem();

    // Set conversion settings
    item.Set("Devmode settings;Resolution", "200");
    item.Set("Save;Output File Format", "TIFF Multipaged");
    item.Set("Save;Append", "0");
    item.Set("Save;Color reduction", "BW");
    item.Set("Save;Dithering method", "Floyd");
    item.Set("TIFF File Format;BW compression", "Group4");
    item.Set("TIFF File Format;Color compression", "LZW RGB");
    item.Set("TIFF File Format;Indexed compression", "LZW");
    item.Set("TIFF File Format;Greyscale compression", "LZW");

    // convert the file, use Microsoft Word or OutsideIn AX
    // whichever is found first
    item.Convert("Microsoft Word;Outside-In AX",
                @"C:\Test\Report.docx",
                @"C:\Test\Out\ConvertedReport");
}
catch (Exception ex)
{
    MessageBox.Show(this, "An error has occurred.\n\n" + ex.ToString());
}
finally
{
    if (item != null)
    {
        System.Runtime.InteropServices.Marshal.FinalReleaseComObject(item);
    }
    item = null;
}
```

**Code Sample - VB**

```
Dim item As PNDocConvQueueServiceLib.IPNDocConvQueueItem

Try
    ' Create the conversion item
    item = New PNDocConvQueueServiceLib.PNDocConvQueueItem()

    ' Set conversion settings
    item.Set("Devmode settings;Resolution", "200")
    item.Set("Save;Output File Format", "TIFF Multipaged")
    item.Set("Save;Append", "0")
    item.Set("Save;Color reduction", "BW")
    item.Set("Save;Dithering method", "Floyd")
    item.Set("TIFF File Format;BW compression", "Group4")
    item.Set("TIFF File Format;Color compression", "LZW RGB")
    item.Set("TIFF File Format;Indexed compression", "LZW")
    item.Set("TIFF File Format;Greyscale compression", "LZW")

    ' convert the file, use Microsoft Word or OutsideIn AX
    ' whichever is found first
    item.Convert("Microsoft Word;Outside-In AX", _
                "C:\Test\Report.docx", _
                "C:\Test\Out\ConvertedReport")

Catch ex As Exception
    MessageBox.Show(Me, "An error has occurred." & vbCrLf & vbCrLf & ex.ToString())

Finally
    If IsNothing(item) Then
        System.Runtime.InteropServices.Marshal.FinalReleaseComObject(item)
        item = Nothing
    End If
End Try
```

Get

Description

Returns the named setting from the settings dictionary collection.

Syntax

```
expression.Get(string Key)
```

where *expression* is an [IPNDocConvQueueItem](#) object.

Returns a **Object** containing the property value. An exception is thrown if that property is not set.

Parameters

String Key

The key to locate in the settings collection.

Exceptions

Exception	Condition
NullReferenceException	The key passed in is null.
COMException	The key does not exist in the collection.

See Also:

[Contains](#) [Set](#) [Remove](#) [Convert](#)

Remove

Description

Removes the setting from the dictionary collection.

Syntax

```
expression.Remove(string Key)
```

where *expression* is an [IPNDocConvQueueItem](#) object.

An exception is thrown if that property is not set.

Parameters

String Key

The key to remove from in the settings collection.

Exceptions

Exception	Condition
NullReferenceException	The key passed in is null.
COMException	The key does not exist in the collection.

See Also:

[Contains](#) [Get](#) [Set](#) [Convert](#)

Set

Description

Adds or replaces (updates) the named setting into settings dictionary collection.

Syntax

```
expression.Set(string Key, object Value)
```

where *expression* is an [IPNDocConvQueueItem](#) object.

Parameters

String Key

The key to add into the settings collection, or the name of an existing key to replace the current one with a new value.

object Value

The object to associate with the key in the settings collection.

Exceptions

Exception	Condition
NullReferenceException	The key passed in is null.

See Also:

[Contains](#) [Get](#) [Remove](#) [Convert](#)

WaitForAvailableConverterPlugIn

Description

Waits until the converter has been loaded by the Document Conversion Service or the time-out interval elapses.

Syntax

expression.WaitForAvailableConverterPlugIn(*ConverterPlugInName*, *Timeout*)

where *expression* is an [IPNDocConvQueueItem](#) object.

Returns a Boolean **True** if the converter was loaded within the specified timeout interval, **False** if it did not.

Parameters

String ConverterPlugInName

The name of the converter.

int Timeout

The length of time, in microseconds, to wait. Passing 0 as the timeout interval will simply test if the converter is loaded and then immediately return. An exception is thrown if the timeout value is passed as INFINITE (-1).

Exceptions

Exception	Condition
COMException	<p>The following errors will cause this exception. The exception message details the cause of the error.</p> <ul style="list-style-type: none"> <i>ConverterPlugInName</i> string is null or empty The <i>Timeout</i> is INFINITE (-1), an infinite timeout value is not allowed.

Exceptions

Exception	Condition
ArgumentNullException	Converter name is null or empty

See Also:

[Contains](#) [Get](#) [Set](#) [Remove](#) [Convert](#) [WaitForRunningDocumentConversionService](#)

WaitForRunningDocumentConversionService

Description

Waits the specified timeout interval for the Document Conversion Service to be running and waiting for documents to convert. Pass a timeout value of 0 to test if the service is running and immediately return.

Syntax

```
expression.WaitForRunningDocumentConversionService(Timeout)
```

where expression is an [IPNDocConvQueueItem](#) object.

Returns a Boolean **True** if the service is running within the specified timeout interval, **False** if it did not.

Parameters

int Timeout

The length of time, in microseconds, to wait. Passing 0 as the timeout interval will simply test if the service is running and then immediately return. An exception is thrown if the timeout value is passed as INFINITE (-1).

Exceptions

Exception	Condition
COMException	The following errors will cause this exception. The exception message details the cause of the error. <ul style="list-style-type: none">The <i>Timeout</i> is INFINITE (-1), an infinite timeout value is not allowed.

See Also:

[Contains](#) [Get](#) [Set](#) [Remove](#) [Convert](#) [WaitForAvailableConverterPlugin](#)

Properties

Errors

Description

Returns an [IPNDocConvQueueItemErrors](#) collection of [IPNDocConvQueueItemError](#) objects. This collection is empty if no errors occurred during conversion. This collection is automatically cleared each time [Convert](#) is called on a new file.

Read-only.

Syntax

expression.Errors


where *expression* is an [IPNDocConvQueueItem](#) object.

Returns an [IPNDocConvQueueItemErrors](#) collection.

See Also:

[Files](#) [Images](#) [Jobs](#) [Messages](#) [Pages](#)

Examples

 **Code Sample - C#**

```
if (item != null)
{
    // Did we have any errors
    PNDocConvQueueServiceLib.IPNDocConvQueueItemErrors errors = item.Errors;
    try
    {
        if (errors.Count > 0)
        {
            // Item had conversion errors
            MessageBox.Show(this, "There were errors during the conversion.");
        }
    }
    finally
    {
        if (errors != null)
        {
            System.Runtime.InteropServices.Marshal.FinalReleaseComObject(errors);
        }
    }
}
```



Code Sample - VB

```
If IsNothing(item) Then

    Dim errors As PNDocConvQueueServiceLib.IPNDocConvQueueItemErrors
    errors = Nothing
    errors = item.Errors

    Try
        If errors.Count > 0 Then
            ' Item had conversion errors
            MessageBox.Show("There were errors during the conversion.")
        End If
    Finally
        If Not IsNothing(errors) Then
            Marshal.FinalReleaseComObject(errors)
        End If
    End Try
End If
```

Files

Description

Returns an [IPNDocConvQueueItemFiles](#) collection of [IPNDocConvQueueItemFile](#) objects for this conversion. Each [IPNDocConvQueueItemFile](#) object represents a file created by the conversion when calling [Convert](#). This collection is automatically cleared each time [Convert](#) is called on a new file.

Read-only.

Syntax

expression.Files

where *expression* is an [IPNDocConvQueueItem](#) object.

Returns an [IPNDocConvQueueItemFiles](#) collection.

See Also:

[Errors](#) [Images](#) [Jobs](#) [Messages](#) [Pages](#)

Examples



Code Sample - C#

```
PNDocConvQueueServiceLib.IPNDocConvQueueItemFiles files = null ;
int itemCount = 0;

try {

    files = item.Files ;
    itemCount = files.Count;

    if (itemCount > 0) {

        foreach (PNDocConvQueueServiceLib.IPNDocConvQueueItemFile file in files) {

            try {
                MessageBox.Show(this, filesList)
            }
            finally {
                System.Runtime.InteropServices.Marshal.FinalReleaseComObject(file);
            }
        }
    }
    else {
        MessageBox.Show(this, "No files were created.");
    }
}
finally
{
    if ( files != null ) {
        System.Runtime.InteropServices.Marshal.FinalReleaseComObject(files);
        files = null ;
    }
}
```



Code Sample - VB

```
Dim itemFileCollection As PNDocConvQueueServiceLib.IPNDocConvQueueItemFiles
Dim itemCount As Integer
itemFileCollection = Nothing

Try
    itemFileCollection = item.Files
    itemCount = itemFileCollection.Count

    If itemCount > 0 Then

        For Each File As PNDocConvQueueServiceLib.IPNDocConvQueueItemFile
            In itemFileCollection
                Try
                    MessageBox.Show(Me, File.FileName)

                    Finally
                        Marshal.FinalReleaseComObject(File)
                    End Try
                Next
            Else
                MessageBox.Show(Me, "No files were created.")
            End If
        Finally
            If Not IsNothing(itemFileCollection) Then
                Marshal.FinalReleaseComObject(itemFileCollection)
            End If
        End Try
```

Images

Description

Returns an [IPNDocConvQueueItemImages](#) collection of [IPNDocConvQueueItemImage](#) objects for this conversion. This collection is automatically cleared each time [Convert](#) is called on a new file.

Read-only.

Syntax

expression.Images


where *expression* is an [IPNDocConvQueueItem](#) object.

Returns an [IPNDocConvQueueItemImages](#) collection.

See Also:

[Errors](#) [Files](#) [Jobs](#) [Messages](#) [Pages](#)

Examples

 **Code Sample - C#**

```
PNDocConvQueueServiceLib.IPNDocConvQueueItemFiles files = null ;
int itemCount = 0;

try {

    files = item.Files ;
    itemCount = files.Count;

    if (itemCount > 0) {

        foreach (PNDocConvQueueServiceLib.IPNDocConvQueueItemFile file in files) {

            try {
                MessageBox.Show(this, filesList)
            }
            finally {
                System.Runtime.InteropServices.Marshal.FinalReleaseComObject(file);
            }
        }
    }
    else {
        MessageBox.Show(this, "No files were created.");
    }
}
finally
{
    if ( files != null ) {
        System.Runtime.InteropServices.Marshal.FinalReleaseComObject(files);
        files = null ;
    }
}
```



Code Sample - VB

```
Dim itemFileCollection As PNDocConvQueueServiceLib.IPNDocConvQueueItemFiles
Dim itemCount As Integer
itemFileCollection = Nothing

Try
    itemFileCollection = item.Files
    itemCount = itemFileCollection.Count

    If itemCount > 0 Then

        For Each File As PNDocConvQueueServiceLib.IPNDocConvQueueItemFile
            In itemFileCollection
                Try
                    MessageBox.Show(Me, File.FileName)

                    Finally
                        Marshal.FinalReleaseComObject(File)
                    End Try
                Next
            Else
                MessageBox.Show(Me, "No files were created.")
            End If
        Finally
            If Not IsNothing(itemFileCollection) Then
                Marshal.FinalReleaseComObject(itemFileCollection)
            End If
        End Try
```


Jobs

Description

Returns an [IPNDocConvQueueItemJobs](#) collection of [IPNDocConvQueueItemJob](#) objects for this conversion. There can be one or more jobs per conversion. This collection is automatically cleared each time [Convert](#) is called on a new file.

Read-only.

Syntax

expression.Jobs

where *expression* is an [IPNDocConvQueueItem](#) object.

Returns an [IPNDocConvQueueItemJobs](#) collection.

See Also:

[Errors](#) [Files](#) [Images](#) [Messages](#) [Pages](#)

Messages

Description

The [IPNDocConvQueueItemMessages](#) collection of [IPNDocConvQueueItemMessage](#) objects for this conversion. This collection contains any information messages reported by the conversion process. This collection is automatically cleared each time [Convert](#) is called on a new file.

These messages are not the same as error messages; see [Errors](#) for the collection of errors.

Read-only.

Syntax

expression.Messages

where *expression* is an [IPNDocConvQueueItem](#) object.

Returns an [IPNDocConvQueueItemMessages](#) collection.

See Also:

[Errors](#) [Files](#) [Images](#) [Jobs](#) [Pages](#)

Pages

Description

Returns an [IPNDocConvQueueItemPages](#) collection of [IPNDocConvQueueItemPage](#) objects for this conversion. This collection is automatically cleared each time [Convert](#) is called on a new file.

Read-only.

Syntax

expression.Pages

where *expression* is an [IPNDocConvQueueItem](#) object.

Returns an [IPNDocConvQueueItemPages](#) collection.

See Also:

[Errors](#) [Files](#) [Images](#) [Jobs](#) [Messages](#)

Events

OnCloseFile

Description

The OnCloseFile event occurs each time a file we are creating is closed. By attaching to this event you can perform custom actions with the newly created file. File information, including page count and file location can be retrieved from the [IPNDocConvQueueItemFile](#) object passed to the event handler.

Syntax

```
objectName_OnCloseFile
```

where the [IPNDocConvQueueItem](#) object is declared with events.

Parameters

IPNDocConvQueueItemFile DocConvQueueItemFile

The [IPNDocConvQueueItemFile](#) object representing the file that was just closed and the conversion job it is part of.

See Also:

[OnEndConversion](#) [OnEndImage](#) [OnEndJob](#) [OnEndPage](#) [OnGetNextOutputFileName](#)
[OnLogMessage](#) [OnOpenFile](#) [OnReportError](#) [OnStartConversion](#) [OnStartImage](#)
[OnStartJob](#) [OnStartPage](#)

OnEndConversion

Description

The OnEndConversion event occurs when the conversion process is complete for the source file.

Syntax

objectName_OnEndConversion

where the [IPNDocConvQueueItem](#) object is declared with events.

See Also:

[OnCloseFile](#) [OnEndImage](#) [OnEndJob](#) [OnEndPage](#) [OnGetNextOutputFileName](#)
[OnLogMessage](#) [OnOpenFile](#) [OnReportError](#) [OnStartConversion](#) [OnStartImage](#)
[OnStartJob](#) [OnStartPage](#)

OnEndImage

Description

The OnEndImage event occurs each time a new image for a page is completed. Currently only one image is created per page.

Syntax

`objectName_OnEndImage`

where the [IPNDocConvQueueItem](#) object is declared with events.

Parameters

IPNDocConvQueueItemImage DocConvQueueItemImage

The [IPNDocConvQueueItemImage](#) object representing the image and the file and job it belongs to.

See Also:

[OnCloseFile](#) [OnEndConversion](#) [OnEndJob](#) [OnEndPage](#) [OnGetNextOutputFileName](#)
[OnLogMessage](#) [OnOpenFile](#) [OnReportError](#) [OnStartConversion](#) [OnStartImage](#)
[OnStartJob](#) [OnStartPage](#)

OnEndJob

Description

The OnEndJob event occurs each time a single print job for a file has completed printing through the converters. For most documents there is only a single print job created when the document is printed, but some applications, like Excel, can send multiple jobs when printing a single file.

Syntax

`objectName_OnEndJob`

where the [IPNDocConvQueueItem](#) object is declared with events.

Parameters

PNDocConvQueueItemJob Job

The [IPNDocConvQueueItemJob](#) object representing one of the print jobs for the file.

See Also:

[OnCloseFile](#) [OnEndConversion](#) [OnEndImage](#) [OnEndPage](#) [OnGetNextOutputFileName](#)
[OnLogMessage](#) [OnOpenFile](#) [OnReportError](#) [OnStartConversion](#) [OnStartImage](#)
[OnStartJob](#) [OnStartPage](#)

OnEndPage

Description

The OnEndPage event occurs each time a new page in an output file is completed.

Syntax

`objectName_OnEndPage`

where the [IPNDocConvQueueItem](#) object is declared with events.

Parameters

PNDocConvQueueItemPage DocConvQueueItemPage

The [IPNDocConvQueueItemPage](#) object representing the page information that was just added to the output file.

See Also:

[OnCloseFile](#) [OnEndConversion](#) [OnEndImage](#) [OnEndJob](#) [OnGetNextOutputFileName](#)

[OnLogMessage](#) [OnOpenFile](#) [OnReportError](#) [OnStartConversion](#) [OnStartImage](#)

[OnStartJob](#) [OnStartPage](#)

OnGetNextOutputFileName

Description

This event rarely needs to be handled; it is easier to pass the desired output file name in the [Convert](#) method.

The OnGetNextOutputFileName event occurs each time a file name needs to be created. Depending on the conversion settings this can happen once for each file if the output is multi-paged or once for every page that is printed if the output is serialized. If you do use this event construct your desired file name and return it as a string value.

Syntax

objectName_OnGetNextOutputFileName

where the [IPNDocConvQueueItem](#) object is declared with events.

Return value is the desired output file name as a string object.

See Also:

[OnCloseFile](#) [OnEndConversion](#) [OnEndImage](#) [OnEndJob](#) [OnEndPage](#)
[OnLogMessage](#) [OnOpenFile](#) [OnReportError](#) [OnStartConversion](#) [OnStartImage](#)
[OnStartJob](#) [OnStartPage](#)

OnLogMessage

Description

The OnLogMessage event occurs when informational messages are logged by the conversion process.

Syntax

objectName_OnLogMessage

where the [IPNDocConvQueueItem](#) object is declared with events.

Parameters

String Msg

The informational message as a string object.

See Also:

[OnCloseFile](#) [OnEndConversion](#) [OnEndImage](#) [OnEndJob](#) [OnEndPage](#)
[OnGetNextOutputFileName](#) [OnOpenFile](#) [OnReportError](#) [OnStartConversion](#)
[OnStartImage](#) [OnStartJob](#) [OnStartPage](#)

OnOpenFile

Description

The OnOpenFile event occurs each time a new output file is opened or created as part of the conversion process. Depending on the conversion settings this can happen once for each file if the output is multi-paged or once for every page that is printed if the output is serialized.

Syntax

```
objectName_OnOpenFile
```

where the [IPNDocConvQueueItem](#) object is declared with events.

Parameters

PNDocConvQueueItemFile DocConvQueueItemFile

The [IPNDocConvQueueItemFile](#) object representing the file on disk currently being created.

See Also:

[OnCloseFile](#) [OnEndConversion](#) [OnEndImage](#) [OnEndJob](#) [OnEndPage](#) [OnGetNextOutputFileName](#)
[OnLogMessage](#) [OnReportError](#) [OnStartConversion](#) [OnStartImage](#) [OnStartJob](#) [OnStartPage](#)

OnReportError

Description

The OnReportError event is fired when an error occurs during the conversion process.

Syntax

`objectName_OnReportError`

where the [IPNDocConvQueueItem](#) object is declared with events.

Parameters

String Error

The error message.

See Also:

[OnCloseFile](#) [OnEndConversion](#) [OnEndImage](#) [OnEndJob](#) [OnEndPage](#) [OnGetNextOutputFileName](#)
[OnLogMessage](#) [OnOpenFile](#) [OnStartConversion](#) [OnStartImage](#) [OnStartJob](#) [OnStartPage](#)

OnStartConversion

Description

The OnStartConversion event occurs when the source file is received to be converted but before the conversion process is started.

Syntax

objectName_OnStartConversion

where the [IPNDocConvQueueItem](#) object is declared with events.

See Also:

[OnCloseFile](#) [OnEndConversion](#) [OnEndImage](#) [OnEndJob](#) [OnEndPage](#) [OnGetNextOutputFileName](#)
[OnLogMessage](#) [OnOpenFile](#) [OnReportError](#) [OnStartImage](#) [OnStartJob](#) [OnStartPage](#)

OnStartImage

Description

The OnStartImage event occurs each time a new image for a page is created. Currently only one image is created per page.

Syntax

`objectName_OnStartImage`

where the [IPNDocConvQueueItem](#) object is declared with events.

Parameters

IPNDocConvQueueItemImage DocConvQueueItemImage

The [IPNDocConvQueueItemImage](#) object representing the image and the file and job it belongs to.

See Also:

[OnCloseFile](#) [OnEndConversion](#) [OnEndImage](#) [OnEndJob](#) [OnEndPage](#) [OnGetNextOutputFileName](#)
[OnLogMessage](#) [OnOpenFile](#) [OnReportError](#) [OnStartConversion](#) [OnStartJob](#) [OnStartPage](#)

OnStartJob

Description

The OnStartJob event occurs each time a single print job for a file has started printing through the converters. For most documents there is only a single print job created when the document is printed, but some applications, like Excel, can send multiple jobs when printing a single file.

Syntax

```
objectName_OnStartJob
```

where the [IPNDocConvQueueItem](#) object is declared with events.

Parameters

IPNDocConvQueueItemJob Job

The IPNDocConvQueueItemJob object representing one of the print jobs for the file.

See Also:

[OnCloseFile](#) [OnEndConversion](#) [OnEndImage](#) [OnEndJob](#) [OnEndPage](#) [OnGetNextOutputFileName](#)
[OnLogMessage](#) [OnOpenFile](#) [OnReportError](#) [OnStartConversion](#) [OnStartImage](#) [OnStartPage](#)

OnStartPage

Description

The OnStartPage event occurs each time a new page in an output file is to be created.

Syntax

`objectName_OnStartPage`

where the [IPNDocConvQueueItem](#) object is declared with events.

Parameters

IPNDocConvQueueItemPage DocConvQueueItemPage

The [IPNDocConvQueueItemPage](#) object representing the page information that will be added to the output file.

See Also:

[OnCloseFile](#) [OnEndConversion](#) [OnEndImage](#) [OnEndJob](#) [OnEndPage](#) [OnGetNextOutputFileName](#)
[OnLogMessage](#) [OnOpenFile](#) [OnReportError](#) [OnStartConversion](#) [OnStartImage](#) [OnStartJob](#)

IPNDocConvQueueItemJobs Collection


Description

A collection of [IPNDocConvQueueItemJob](#) objects. There can be one or more jobs created per conversion.


Each job object represents

This collection is automatically cleared each time [Convert](#) is called on a new file.

Public Methods

 Item	Returns an individual IPNDocConvQueueItemJob item from the collection. The collection can be indexed by position or by JobGUID string.
--	--

Public Properties

 Count	Read-only; Returns an Integer that represents the number of IPNDocConvQueueItemJob objects in the collection.
---	---

Methods

Item

Description

Returns an individual [IPNDocConvQueueItemJob](#) item from the collection.

Syntax

expression.Item(*Index*)

where *expression* is an [IPNDocConvQueueItemJobs](#) collection

Returns an [IPNDocConvQueueItemJob](#) item, or E_FAIL if the *Index* is out of range.

Parameters

Object Index

The collection can be indexed by integer position by integer position or by [JobGUID](#) string.

See Also:

[Count IPNDocConvQueueItemJob](#)

Properties

Count

Description

Returns the number of [IPNDocConvQueueItemJob](#) objects in the [IPNDocConvQueueItemJobs](#) collection.

Read-only Integer.

Syntax

expression.Count

where *expression* is an [IPNDocConvQueueItemJobs](#) collection

Returns an **Integer**.

See Also:

[Item IPNDocConvQueueItemJob](#)

IPNDocConvQueueItemJob

Description









Each IPNDocConvQueueItemJob object represents a single print job created when calling the [Convert](#) function.



Many of the converters used by Document Conversion Service will use the Document Conversion Service 3.0 printer to do the conversion. For most documents there is only a single print job created when the document is printed, but some applications can send multiple jobs when printing a single file. One example of this is Excel when printing a workbook containing multiple worksheets at different print quality settings. Excel will create a separate print job for each group of worksheets with different print qualities.

The job object is identified by a unique identifier, the [JobGUID](#) and contains information about the job such as the job status and the number of pages spooled and printed. It also provides access to two collections:

- the [IPNDocConvQueueItemFiles](#) collection of the files created by this job
- the [IPNDocConvQueueItemPages](#) collection of the printed pages belonging to this job

Public Properties

 BytesPrinted	Read-only; How much of the document has been printed.
 BytesSpooled	Read-only; Size of the document (in bytes)in the printer queue.
 Document	Read-only; Name of the document printed.
 Files	Read-only; IPNDocConvQueueItemFiles collection of IPNDocConvQueueItemFile objects created by this print job.
 JobGUID	Read-only; Unique identifier for this object. Can be used to query the IPNDocConvQueueItemJobs collection.
 JobID	Read-only; non-unique identifier used by the Windows printing sub-system.
 Pages	Read-only; collection of IPNDocConvQueueItemPage objects created by this print job.
 PagesPrinted	Read-only; count of the number of pages printed.
 PagesSpooled	Read-only; count of the number of pages spooled.
 Status	Read-only; current print status of the job as an Integer value.
 StatusMessage	Read-only; current print status of the job as a string value.

 SubmittedTime	Read-only; VT_DATE variant that specifies the time this document was spooled.
 UserName	Read-only; name of the user who printed the document.

Properties

BytesPrinted

Description

Returns the size of the printed job in bytes. This can be different from [BytesSpooled](#).

Read-only.

Syntax

expression.BytesPrinted

where *expression* is an [IPNDocConvQueueItemJob](#) object.

Returns an **Integer**.

See Also:

[BytesSpooled](#) [Document Files](#) [JobGUID](#) [JobID](#) [Pages](#) [PagesPrinted](#) [PagesSpooled](#) [Status](#)
[StatusMessage](#) [SubmittedTime](#) [UserName](#)

BytesSpooled

Description

The size of the spooled job in bytes.

Read-only.

Syntax

expression.BytesSpooled

where *expression* is an [IPNDocConvQueueItemJob](#) object.

Returns an **Integer**.

See Also:

[BytesPrinted](#) [Document](#) [Files](#) [JobGUID](#) [JobID](#) [Pages](#) [PagesPrinted](#) [PagesSpooled](#) [Status](#)
[StatusMessage](#) [SubmittedTime](#) [UserName](#)

Document

Description

The name of the document printed that created this print job. This is the name the printing application uses in the print queue. It can be different from the actual document name.

Read-only.

Syntax

expression.Document

where *expression* is an [IPNDocConvQueueItemJob](#) object.

Returns a **String**.

See Also:

[BytesPrinted](#) [BytesSpooled](#) [Files](#) [JobGUID](#) [JobID](#) [Pages](#) [PagesPrinted](#) [PagesSpooled](#) [Status](#)
[StatusMessage](#) [SubmittedTime](#) [UserName](#)

Files

Description

The [IPNDocConvQueueItemFiles](#) collection of [IPNDocConvQueueItemFile](#) objects for this job. There will be one [IPNDocConvQueueItemFile](#) object for every file created by this job.

Read-only.

Syntax

expression.Files

where *expression* is an [IPNDocConvQueueItemJob](#) object.

Returns an [IPNDocConvQueueItemFiles](#) collection.

See Also:

[BytesPrinted](#) [BytesSpooled](#) [Document](#) [JobGUID](#) [JobID](#) [Pages](#) [PagesPrinted](#) [PagesSpooled](#) [Status](#)
[StatusMessage](#) [SubmittedTime](#) [UserName](#)

JobGUID

Description

A string based unique identifier for this object. This string can be used to query the [IPNDocConvQueueItemJobs](#) collection.

Read-only.

Syntax

expression.JobGUID

where *expression* is an [IPNDocConvQueueItemJob](#) object.

Returns a **String**.

See Also:

[BytesPrinted](#) [BytesSpooled](#) [Document Files](#) [JobID](#) [Pages](#) [PagesPrinted](#) [PagesSpooled](#) [Status](#)
[StatusMessage](#) [SubmittedTime](#) [UserName](#)

JobID

Description

This is a non-unique numerical identifier used by the Windows printing sub-system.

Read-only.

Syntax

expression.JobID

where *expression* is an [IPNDocConvQueueItemJob](#) object.

Returns an **Integer**.

See Also:

[BytesPrinted](#) [BytesSpooled](#) [Document](#) [Files](#) [JobGUID](#) [Pages](#) [PagesPrinted](#) [PagesSpooled](#) [Status](#)
[StatusMessage](#) [SubmittedTime](#) [UserName](#)

Pages

Description

The [IPNDocConvQueueItemPages](#) collection of [IPNDocConvQueueItemPage](#) objects for this job. There will be one [IPNDocConvQueueItemPage](#) object for every document page printed by this job.

Read-only.

Syntax

expression.Pages

where *expression* is a [IPNDocConvQueueItemJob](#) object.

Returns an [IPNDocConvQueueItemPages](#) collection.

See Also:

[BytesPrinted](#) [BytesSpooled](#) [Document Files](#) [JobGUID](#) [JobID](#) [PagesPrinted](#) [PagesSpooled](#) [Status](#)
[StatusMessage](#) [SubmittedTime](#) [UserName](#)

PagesPrinted

Description

Returns the number of pages printed. This can be different from [PagesSpooled](#).

Read-only.

Syntax

expression.PagesPrinted

where *expression* is an [IPNDocConvQueueItemJob](#) object.

Returns an **Integer**.

See Also:

[BytesPrinted](#) [BytesSpooled](#) [Document](#) [Files](#) [JobGUID](#) [JobID](#) [Pages](#) [PagesSpooled](#) [Status](#)
[StatusMessage](#) [SubmittedTime](#) [UserName](#)

PagesSpooled

Description

Returns the number of pages spooled. This can be different from [PagesPrinted](#).

Read-only.

Syntax

expression.PagesSpooled

where *expression* is an [IPNDocConvQueueItemJob](#) object.

Returns an **Integer**.

See Also:

[BytesPrinted](#) [BytesSpooled](#) [Document](#) [Files](#) [JobGUID](#) [JobID](#) [Pages](#) [PagesPrinted](#) [Status](#)
[StatusMessage](#) [SubmittedTime](#) [UserName](#)

Status

Description

The print status of the job as a numerical value. See the **Remarks** section for a list of the status values and what they mean.

Read-only.

Syntax

expression.Status

where *expression* is an [IPNDocConvQueueItemJob](#) object.

Returns an **Integer**.

Remarks

The status can be one or more of the values in the table below. These are the same values used by the *JOB_INFO_2* structure in Microsoft's Win32 Printing and Print Spooler functions and structures. See the Microsoft documentation for more details.

The values are OR'd together to define the current status of the job. To determine which values, the hexadecimal values must be examined:

If Status = 388, which is 0x00000184

```
JOB_STATUS_DELETED      0x00000100
JOB_STATUS_PRINTED      0x00000080
JOB_STATUS_DELETING     0x00000004
-----
                          0x00000184
```

Job Status	Hexadecimal Value	Integer Value
JOB_STATUS_PAUSED	0x00000001	1
JOB_STATUS_ERROR	0x00000002	2
JOB_STATUS_DELETING	0x00000004	4
JOB_STATUS_SPOOLING	0x00000008	8
JOB_STATUS_PRINTING	0x00000010	16
JOB_STATUS_OFFLINE	0x00000020	32
JOB_STATUS_PAPEROUT	0x00000040	64
JOB_STATUS_PRINTED	0x00000080	128
JOB_STATUS_DELETED	0x00000100	256
JOB_STATUS_BLOCKED_DEVQ	0x00000200	512
JOB_STATUS_USER_INTERVENTION	0x00000400	1024
JOB_STATUS_RESTART	0x00000800	2048
JOB_STATUS_COMPLETE	0x00001000	4096
JOB_STATUS_RETAINED	0x00002000	8192
JOB_STATUS_RENDERING_LOCALLY	0x00004000	16384

See Also:

[BytesPrinted](#) [BytesSpooled](#) [Document Files](#) [JobGUID](#) [JobID](#) [Pages](#) [PagesPrinted](#) [PagesSpooled](#) [StatusMessage](#) [SubmittedTime](#) [UserName](#)

StatusMessage

Description

The current print status of the job as a string value. This value can be an empty string.

Read-only.

Syntax

expression.StatusMessage

where *expression* is an [IPNDocConvQueueItemJob](#) object.

Returns a **String**.

See Also:

[BytesPrinted](#) [BytesSpooled](#) [Document](#) [Files](#) [JobGUID](#) [JobID](#) [Pages](#) [PagesPrinted](#) [PagesSpooled](#) [Status](#) [SubmittedTime](#) [UserName](#)

SubmittedTime

Description

Returns the time this document was spooled.

Read-only.

Syntax

expression.SubmittedTime

where *expression* is an [IPNDocConvQueueItemJob](#) object.

Returns a VT_DATE **VARIANT**.

See Also:

[BytesPrinted](#) [BytesSpooled](#) [Document](#) [Files](#) [JobGUID](#) [JobID](#) [Pages](#) [PagesPrinted](#) [PagesSpooled](#)
[Status](#) [StatusMessage](#) [UserName](#)

UserName

Description

Returns the name of the user who printed the document.

Read-only.

Syntax

expression.UserName

where *expression* is an [IPNDocConvQueueItemJob](#) object.

Returns a **String**.

See Also:


[BytesPrinted](#) [BytesSpooled](#) [Document](#) [Files](#) [JobGUID](#) [JobID](#) [Pages](#) [PagesPrinted](#) [PagesSpooled](#) [Status](#) [StatusMessage](#) [SubmittedTime](#)

IPNDocConvQueueItemFiles Collection


Description

A collection of [IPNDocConvQueueItemFile](#) objects. Allows quick iteration through the collection using the `foreach` statement of the C# language and the `for each` statement in Visual Basic.

Public Methods

 Item	Returns an individual IPNDocConvQueueItemFile item from the collection. The collection is indexed by position.
--	--

Public Properties

 Count	Returns an Integer that represents the number of IPNDocConvQueueItemFile objects in the collection.
---	---

Methods

Item

Description

Returns an individual [IPNDocConvQueueItemFile](#) item from the collection.

Syntax

expression.Item(*Index*)

where *expression* is an [IPNDocConvQueueItemFiles](#) collection

Returns an [IPNDocConvQueueItemFile](#) item, or E_FAIL if the *Index* is out of range.

Parameters

Object Index

The collection is indexed by integer position.

See Also:

[Count IPNDocConvQueueItemFile](#)

Properties

Count

Description

Returns the number of IPNDocConvQueueItemFile objects in the [IPNDocConvQueueItemFiles](#) collection. Each time a new output file is created, an IPNDocConvQueueItemFile object is added to the collection. The output settings for file creation, such as multipage or serialized output, and append mode will determine the resulting number of IPNDocConvQueueItemFile objects in the collection.

Read-only.

Syntax

expression.Count

where *expression* is an [IPNFiles](#) collection

Returns an **Integer**.

See Also:





[Item IPNDocConvQueueItemFile](#)

IPNDocConvQueueItemFile

Description

An IPNDocConvQueueItemFile object is created for every file created. The file object represents the physical file on disk and provides access to the full output filename and two collections: an [IPNDocConvQueueItemPages](#) collection of the printed pages used to create the file and an [IPNDocConvQueueItemImages](#) collection of the images that make up each page of the file.

Public Properties

 Filename	Read-only; The filename of the file created.
 Images	Read-only; IPNDocConvQueueItemImages collection of IPNDocConvQueueItemImage objects in this file. This value is updated during the printing process.
 Job	Read-only; The parent IPNDocConvQueueItemJob object that created this IPNFile object.
 Pages	Read-only; IPNDocConvQueueItemPages collection of IPNDocConvQueueItemPage objects in this file. This value is updated during the printing process.

Properties

Filename

Description

The name of the file created. This is the fully qualified path, including directory and filename.

Read-only.

Syntax

expression.Filename

where *expression* is an [IPNDocConvQueueItemFile](#) object

Returns a **String**.

See Also:

[Images](#) [Job](#) [Pages](#)

Images

Description

The [IPNDocConvQueueItemImages](#) collection of [IPNDocConvQueueItemImage](#) objects for this file. There will be one [IPNDocConvQueueItemImage](#) object for every converted page.

Read-only.

Syntax

expression.Images

where *expression* is an [IPNDocConvQueueItemImage](#) object

Returns an [IPNDocConvQueueItemImages](#) collection.

See Also:

[Filename Job Pages](#)

Job

Description

The parent [IPNDocConvQueueItemJob](#) object that created this IPNDocConvQueueItemFile object.

Read-only.

Syntax

expression.Job

where *expression* is an [IPNDocConvQueueItemFile](#) object

Returns an [IPNDocConvQueueItemJob](#) object.

See Also:

[Filename](#) [Images](#) [Pages](#)

Pages

Description

The [IPNDocConvQueueItemPages](#) collection of [IPNDocConvQueueItemPage](#) objects for this file. There will be one [IPNDocConvQueueItemPage](#) object for every page that was put in this file.

Read-only.

Syntax

expression.Pages

where *expression* is an [IPNDocConvQueueItemFile](#) object

Returns an [IPNDocConvQueueItemPages](#) collection.

See Also:


[Filename Images Job](#)

IPNDocConvQueueItemPages Collection


Description

A collection of [IPNDocConvQueueItemPage](#) objects. Allows quick iteration through the collection using the `foreach` statement of the C# language and the `for each` statement in Visual Basic.

Public Methods

 Item	Returns an individual IPNDocConvQueueItemPage item from the collection. The collection is indexed by position.
--	--

Public Properties

 Count	Returns an Integer that represents the number of IPNDocConvQueueItemPage objects in the collection.
---	---

Methods

Item

Description

Returns an individual [IPNDocConvQueueItemPage](#) item from the collection.

Syntax

expression.Item(*Index*)

where *expression* is an [IPNDocConvQueueItemPages](#) collection

Returns an [IPNDocConvQueueItemPage](#) item, or E_FAIL if the *Index* is out of range.

Parameters

Object Index

The collection is indexed by integer position.

See Also:

[Count IPNDocConvQueueItemPage](#)

Properties

Count

Description

Returns the number of [IPNDocConvQueueItemPage](#) objects in the [IPNDocConvQueueItemPages](#) collection.

Read-only.

Syntax

expression.Count

where *expression* is an [IPNDocConvQueueItemPages](#) collection

Returns an **Integer**.

See Also:

[Item IPNDocConvQueueItemPage](#)












IPNDocConvQueueItemPage

Description

An IPNPage object is created for every page of the document or file. The page object represents the print settings of the page when spooled to the Document Conversion Service printer. These settings are different from the [IPNDocConvQueueItemImage](#) settings, which are the settings of the output file created. For instance, printing a single page document in color and creating a fax resolution TIFF image will give an IPNDocConvQueueItemPage object with a *BitsPerPixel = 24*, and an IPNDocConvQueueItemImage object with *BitsPerPixel = 1*.

The page object also provides access to two collections: an [IPNDocConvQueueItemFiles](#) collection of files that contain this page as an [IPNDocConvQueueItemImage](#), currently only a collection of one, and an [IPNDocConvQueueItemImages](#) collection of the images that were created from this page, also currently only a collection of one.

Public Properties

 BitsPerPixel	Read-only; The bits per pixel, or color depth of the printed page.
 Files	Read-only; IPNDocConvQueueItemFiles collection of IPNDocConvQueueItemFile objects created by this page. Currently each page belongs to a single file.
 HeightInPixels	Read-only; The height of the printed page in pixels.
 Images	Read-only; IPNDocConvQueueItemImages collection of IPNDocConvQueueItemImage objects in this page. Currently each page creates a single image.
 Job	Read-only; The parent IPNDocConvQueueItemJob object that created this IPNDocConvQueueItemPage object.
 Orientation	Read-only; The orientation of the page, either <i>Portrait</i> or <i>Landscape</i> .
 PageNumber	Read-only; The page number of the page.
 Skipped	Read-only; Boolean value True if the page was skipped.
 WidthInPixels	Read-only; The weight of the printed page in pixels.
 XPixelsPerInch	Read-only; The vertical dots per inch, or resolution, of the page.
 YPixelsPerInch	Read-only; The horizontal dots per inch, or resolution, of the page.

Properties

BitsPerPixel

Description

This is the color depth, or bit depth of the page. This can be different from the [BitsPerPixel](#) values in any [IPNDocConvQueueItemImage](#) objects in the [Images](#) collection. It is commonly 1 for black and white, or monochrome printing, and 24 when printing in color.

Read-only.

Syntax

expression.BitsPerPixel

where *expression* is an [IPNPage](#) object

Returns an **Integer**.

See Also:

[Files](#) [HeightInPixels](#) [Images](#) [Job](#) [Orientation](#) [PageNumber](#) [Skipped](#) [WidthInPixels](#) [XPixelsPerInch](#) [YPixelsPerInch](#)

Files

Description

The [IPNDocConvQueueItemFiles](#) collection of [IPNDocConvQueueItemFile](#) objects for this page. There will be one [IPNDocConvQueueItemFile](#) object for every page.

Read-only.

Syntax

expression.Files

where *expression* is an [IPNPage](#) object

Returns an [IPNFiles](#) collection.

See Also:

[BitsPerPixel](#) [HeightInPixelsImages](#) [Job](#) [Orientation](#) [PageNumber](#) [Skipped](#) [WidthInPixels](#)
[XPixelsPerInch](#) [YPixelsPerInch](#)

HeightInPixels

Description

This is the height of the page in pixels.

Read-only.

Syntax

expression.HeightInPixels

where *expression* is an [IPNDocConvQueueItemPage](#) object

Returns an **Integer**.

See Also:

[BitsPerPixel](#) [Files](#) [Images](#) [Job](#) [Orientation](#) [PageNumber](#) [Skipped](#) [WidthInPixels](#) [XPixelsPerInch](#) [YPixelsPerInch](#)

Images

Description

The [IPNDocConvQueueItemImages](#) collection of [IPNDocConvQueueItemImage](#) objects for this page. There will be one [IPNDocConvQueueItemImage](#) object for every page.

Read-only.

Syntax

expression.Images

where *expression* is an [IPNDocConvQueueItemPage](#) object

Returns an [IPNDocConvQueueItemImages](#) collection.

See Also:

[BitsPerPixel](#) [Files](#) [HeightInPixels](#) [Job](#) [Orientation](#) [PageNumber](#) [Skipped](#) [WidthInPixels](#) [XPixelsPerInch](#) [YPixelsPerInch](#)

Job

Description

The parent [IPNDocConvQueueItemJob](#) object that created this IPNDocConvQueueItemPage object.

Read-only.

Syntax

expression.Job

where *expression* is an [IPNDocConvQueueItemPage](#) object

Returns an [IPNDocConvQueueItemJob](#) object.

See Also:

[BitsPerPixel](#) [Files](#) [HeightInPixels](#) [Images](#) [Orientation](#) [PageNumber](#) [Skipped](#) [WidthInPixels](#)
[XPixelsPerInch](#) [YPixelsPerInch](#)

Orientation

Description

This is the orientation, either *Portrait* or *Landscape*, of the page.

Read-only.

Syntax

`expression.Orientation`

where `expression` is an [IPNDocConvQueueItemPage](#) object

Returns an **Integer** where Portrait = 0 and Landscape = 1.

See Also:

[BitsPerPixel](#) [Files](#) [HeightInPixels](#) [Images](#) [Job](#) [PageNumber](#) [Skipped](#) [WidthInPixels](#) [XPixelsPerInch](#)
[YPixelsPerInch](#)

PageNumber

Description

This is the page number of the page. When appending, this can be different from the page number of the image in the resulting file.

Read-only.

Syntax

expression.PageNumber

where *expression* is an [IPNDocConvQueueItemPage](#) object

Returns an **Integer**.

See Also:

[BitsPerPixel](#) [Files](#) [HeightInPixels](#) [Images](#) [Job](#) [Orientation](#) [Skipped](#) [WidthInPixels](#) [XPixelsPerInch](#) [YPixelsPerInch](#)

Skipped

Description

This Boolean property is True if the page was skipped.

Read-only.

Syntax

expression.Skipped

where *expression* is an [IPNDocConvQueueItemPage](#) object

Returns a **Boolean**.

See Also:

[BitsPerPixel](#) [Files](#) [HeightInPixels](#) [Images](#) [Job](#) [Orientation](#) [PageNumber](#) [WidthInPixels](#) [XPixelsPerInch](#) [YPixelsPerInch](#)

WidthInPixels

Description

This is the width of the page in pixels.

Read-only.

Syntax

expression.WidthInPixels

where *expression* is an [IPNDocConvQueueItemPage](#) object

Returns an **Integer**.

See Also:

[BitsPerPixel](#) [Files](#) [HeightInPixels](#) [Images](#) [Job](#) [Orientation](#) [PageNumber](#) [Skipped](#) [XPixelsPerInch](#)
[YPixelsPerInch](#)

XPixelsPerInch

Description

This is the vertical dots per inch (DPI), or resolution, of the page.

Read-only.

Syntax

expression.XPixelsPerInch

where *expression* is an [IPNDocConvQueueItemPage](#) object

Returns an **Integer**.

See Also:

[BitsPerPixel](#) [Files](#) [HeightInPixels](#) [Images](#) [Job](#) [Orientation](#) [PageNumber](#) [Skipped](#) [WidthInPixels](#)
[YPixelsPerInch](#)

YPixelsPerInch

Description

This is the horizontal dots per inch, or resolution, of the page.

Read-only.

Syntax

expression.YPixelsPerInch

where *expression* is an [IPNDocConvQueueItemPage](#) object

Returns an **Integer**.

See Also:


[BitsPerPixel](#) [Files](#) [HeightInPixels](#) [Images](#) [Job](#) [Orientation](#) [PageNumber](#) [Skipped](#) [WidthInPixels](#)
[XPixelsPerInch](#)

IPNDocConvQueueItemImages Collection


Description

A collection of [IPNDocConvQueueItemImage](#) objects. Allows quick iteration through the collection using the `foreach` statement of the C# language and the `for each` statement in Visual Basic.

Public Methods

 Item	Returns an individual IPNDocConvQueueItemImage item from the collection. The collection is indexed by position.
--	---

Public Properties

 Count	Returns an Integer that represents the number of IPNDocConvQueueItemImage objects in the collection.
---	--

Methods

Item

Description

Returns an individual [IPNDocConvQueueItemImage](#) item from the collection.

Syntax

expression.Item(*Index*)

where *expression* is an [IPNDocConvQueueItemImages](#) collection

Returns an [IPNDocConvQueueItemImage](#) item, or E_FAIL if the *Index* is out of range.

Parameters

Object Index

The collection is indexed by integer position.

See Also:

[Count IPNDocConvQueueItemImage](#)

Properties

Count

Description

Returns the number of IPNDocConvQueueItemImage objects in the [IPNDocConvQueueItemImages](#) collection.

Read-only.

Syntax

expression.Count

where *expression* is an [IPNDocConvQueueItemImages](#) collection

Returns an **Integer**.

See Also:

[Item IPNDocConvQueueItemImage](#)









IPNDocConvQueueItemImage

Description

An image object represents a single page in the output physical file on disk.

The image object also provides access to two collections: an [IPNDocConvQueueItemFiles](#) collection of files that contain this image, currently only a collection of one, and an [IPNDocConvQueueItemImages](#) collection of the pages that were used to create this image, also currently only a collection of one.

Public Properties

 BitsPerPixel	Read-only; The bits per pixel, or color depth of the output image.
 Files	Read-only; IPNDocConvQueueItemFiles collection of IPNDocConvQueueItemFile that reference this image. Currently each image belongs to a single file. This value is updated during the printing process.
 HeightInPixels	Read-only; The height of the output image in pixels.
 Job	Read-only; The parent IPNDocConvQueueItemJob object that created this IPNFile object.
 Orientation	Read-only; The orientation of the output image, either <i>Portrait</i> or <i>Landscape</i> .
 PageNumber	Read-only; The page number of the image in the output file.
 Pages	Read-only; IPNDocConvQueueItemPages collection of IPNDocConvQueueItemPage objects in the output file. This value is updated during the printing process.
 RotationInDegress	Read-only; The rotation of the image in the output file.
 WidthInPixels	Read-only; The weight of the image in pixels.
 XPixelsPerInch	Read-only; The vertical dots per inch, or resolution, of the output image.
 YPixelsPerInch	Read-only; The horizontal dots per inch, or resolution, of the output image.

Properties

BitsPerPixel

Description

This is the color depth, or bit depth of the output image. This can be different from the BitsPerPixel values in any [IPNDocConvQueueItemPage](#) objects in the [Pages](#) collection. This value depends on the output settings passed used by [IPNDocConvQueueItem](#).

Read-only.

Syntax

expression.BitsPerPixel

where *expression* is an [IPNDocConvQueueItemImage](#) object

Returns an **Integer**.

See Also:

[Files](#) [HeightInPixels](#) [Job](#) [Orientation](#) [PageNumber](#) [Pages](#) [RotationInDegrees](#) [WidthInPixels](#)
[XPixelsPerInch](#) [YPixelsPerInch](#)

File

Description

The [IPNDocConvQueueItemFile](#) object for this image.

Read-only.

Syntax

expression.Files

where *expression* is an [IPNDocConvQueueItemImage](#) object

Returns an [IPNDocConvQueueItemFile](#) object.

See Also:

[BitsPerPixel](#) [HeightInPixels](#) [Job](#) [Orientation](#) [PageNumber](#) [Pages](#) [RotationInDegrees](#) [WidthInPixels](#)
[XPixelsPerInch](#) [YPixelsPerInch](#)

HeightInPixels

Description

This is the height of the output image in pixels.

Read-only.

Syntax

expression.HeightInPixels

where *expression* is an [IPNDocConvQueueItemImage](#) object

Returns an **Integer**.

See Also:

[BitsPerPixel](#) [Files](#) [Job](#) [Orientation](#) [PageNumber](#) [Pages](#) [RotationInDegrees](#) [WidthInPixels](#)
[XPixelsPerInch](#) [YPixelsPerInch](#)

Job

Description

The parent [IPNDocConvQueueItemJob](#) object that created this IPNDocConvQueueItemImage object.

Read-only.

Syntax

expression.Job

where *expression* is an [IPNDocConvQueueItemImage](#) object

Returns an [IPNDocConvQueueItemJob](#) object.

See Also:

[BitsPerPixel](#) [Files](#) [HeightInPixels](#) [Orientation](#) [PageNumber](#) [Pages](#) [RotationInDegrees](#) [WidthInPixels](#)
[XPixelsPerInch](#) [YPixelsPerInch](#)

Orientation

Description

This is the orientation, either *Portrait* or *Landscape*, of the output image.

Read-only.

Syntax

`expression.Orientation`

where `expression` is an [IPNDocConvQueueItemImage](#) object

Returns an **Integer** where Portrait = 0 and Landscape = 1.

See Also:

[BitsPerPixel](#) [Files](#) [HeightInPixels](#) [Job](#) [PageNumber](#) [Pages](#) [RotationInDegrees](#) [WidthInPixels](#)
[XPixelsPerInch](#) [YPixelsPerInch](#)

PageNumber

Description

This is the page number of the image in the output file. When appending or creating serialized output, this can be different from the page number of the original printed page.

Read-only.

Syntax

expression.PageNumber

where *expression* is an [IPNDocConvQueueItemImage](#) object

Returns an **Integer**.

See Also:

[BitsPerPixel](#) [Files](#) [HeightInPixels](#) [Job](#) [Orientation](#) [Pages](#) [RotationInDegrees](#) [WidthInPixels](#)
[XPixelsPerInch](#) [YPixelsPerInch](#)

Pages

Description

The [IPNDocConvQueueItemPages](#) collection of [IPNDocConvQueueItemPage](#) objects used to create this output image.

Read-only.

Syntax

expression.Pages

where *expression* is an [IPNDocConvQueueItemImage](#) object

Returns an [IPNDocConvQueueItemPages](#) collection.

See Also:

[BitsPerPixel](#) [Files](#) [HeightInPixels](#) [Job](#) [Orientation](#) [PageNumber](#) [RotationInDegrees](#) [WidthInPixels](#)
[XPixelsPerInch](#) [YPixelsPerInch](#)

RotationInDegrees

Description

This is the rotation, one of 0° , 90° , 180° , 270° , of the output image. The output images are always rotated counter-clockwise.

Read-only.

Syntax

expression.BitsPerPixel

where *expression* is an [IPNDocConvQueueItemImage](#) object

Returns an **Integer**, one of 0, 90, 180 or 270.

See Also:

[BitsPerPixel](#) [Files](#) [HeightInPixels](#) [Job](#) [Orientation](#) [PageNumber](#) [Pages](#) [WidthInPixels](#) [XPixelsPerInch](#) [YPixelsPerInch](#)

WidthInPixels

Description

This is the width of the output image in pixels.

Read-only.

Syntax

expression.WidthInPixels

where *expression* is an [IPNDocConvQueueItemImage](#) object

Returns an **Integer**.

See Also:

[BitsPerPixel](#) [Files](#) [HeightInPixels](#) [Job](#) [Orientation](#) [PageNumber](#) [Pages](#) [RotationInDegrees](#)
[XPixelsPerInch](#) [YPixelsPerInch](#)

XPixelsPerInch

Description

This is the vertical dots per inch (DPI), or resolution, of the output image.

Read-only.

Syntax

expression.XPixelsPerInch

where *expression* is an [IPNDocConvQueueItemImage](#) object

Returns an **Integer**.

See Also:

[BitsPerPixel](#) [Files](#) [HeightInPixels](#) [Job](#) [Orientation](#) [PageNumber](#) [Pages](#) [RotationInDegrees](#)
[WidthInPixels](#) [YPixelsPerInch](#)

YPixelsPerInch

Description

This is the horizontal dots per inch, or resolution, of the output image.

Read-only.

Syntax

expression.YPixelsPerInch

where *expression* is an [IPNDocConvQueueItemImage](#) object

Returns an **Integer**.

See Also:


[BitsPerPixel](#) [Files](#) [HeightInPixels](#) [Job](#) [Orientation](#) [PageNumber](#) [Pages](#) [RotationInDegrees](#)
[WidthInPixels](#) [XPixelsPerInch](#)

IPNDocConvQueueItemMessages Collection


Description

A collection of [IPNDocConvQueueItemMessage](#) objects. Each message object wraps a string message that contains information from the conversion process when calling [Convert](#). Allows quick iteration through the collection using the `foreach` statement of the C# language and the `for each` statement in Visual Basic.

Public Methods

 Item	Returns an individual IPNDocConvQueueItemMessage item from the collection. The collection is indexed by position.
--	---

Public Properties

 Count	Returns an Integer that represents the number of IPNDocConvQueueItemMessage objects in the collection.
---	--

Methods

Item

Description

Returns an individual [IPNDocConvQueueItemMessage](#) item from the collection.

Syntax

expression.Item(*Index*)

where *expression* is an [IPNDocConvQueueItemMessages](#) collection

Returns an [IPNDocConvQueueItemMessage](#) item, or E_FAIL if the *Index* is out of range.

Parameters

Object Index

The collection is indexed by integer position.

See Also:

[Count IPNDocConvQueueItemMessage](#)

Properties

Count

Description

Returns the number of IPNDocConvQueueMessage objects in the [IPNDocConvQueueItemMessages](#) collection.

Read-only.

Syntax

expression.Count

where *expression* is an [IPNDocConvQueueItemMessages](#) collection

Returns an **Integer**.

See Also:


[Item IPNDocConvQueueItemMessage](#)

IPNDocConvQueueItemMessage

Description

This object wraps a string message that contains information from the conversion process.

Public Properties

 Value	Read-only; The message as a String.
---	-------------------------------------

Properties

Value

Description

This is the message content.

Read-only.

Syntax

expression.Value

where *expression* is an [IPNDocConvQueueItemMessage](#) object.

Returns an **String**.


IPNDocConvQueueItemErrors Collection

Description


A collection of [IPNDocConvQueueItemError](#) objects. This collection will be empty if the conversion succeeds.

Allows quick iteration through the collection using the `foreach` statement of the C# language and the `for each` statement in Visual Basic.

Public Methods

 Item	Returns an individual IPNDocConvQueueItemError item from the collection. The collection is indexed by position.
--	---

Public Properties

 Count	Returns an Integer that represents the number of IPNDocConvQueueItemError objects in the collection.
---	--

Methods

Item

Description

Returns an individual [IPNDocConvQueueItemError](#) item from the collection.

Syntax

expression.Item(Index)

where *expression* is an [IPNDocConvQueueItemErrors](#) collection

Returns an [IPNDocConvQueueItemError](#) item, or E_FAIL if the *Index* is out of range.

Parameters

Object Index

The collection is indexed by integer position.

See Also:

[Count IPNDocConvQueueItemError](#)

Properties

Count

Description

Returns the number of IPNDocConvQueueItemError objects in the [IPNDocConvQueueItemErrors](#) collection.

Read-only.

Syntax

expression.Count

where *expression* is an [IPNDocConvQueueItemErrors](#) collection

Returns an **Integer**.

See Also:


[Item IPNDocConvQueueItemError](#)

IPNDocConvQueueItemError

Description

An object represents a single error message.

Public Properties

 Value	Read-only; The error message as a String .
---	---

Properties

Value

Description

This is the error message content.

Read-only.

Syntax

expression.Value

where *expression* is an [IPNDocConvQueueItemError](#) object.

Returns an **String**.