

# **PDF Converter Services - Installation & Administration Guide**

Muhimbi Ltd

**Version 6.1**



## Document Control

Draft	Author	Date	Comment
3.0	Muhimbi	12/11/2009	Completely revised for version 3.
3.1	Muhimbi	06/01/2010	Updated for version 3.1
3.2	Muhimbi	23/02/2010	Updated for version 3.2
3.4	Muhimbi	29/03/2010	Revised for PDF Converter Services
3.5	Muhimbi	26/05/2010	Revised for version 3.5
4.0	Muhimbi	08/09/2010	Updated for version 4.0
4.1	Muhimbi	09/11/2010	Updated for version 4.1
5.0	Muhimbi	20/04/2011	Updated for version 5.0
5.1	Muhimbi	02/09/2011	Updated for version 5.1
5.2	Muhimbi	12/01/2012	Updated for version 5.2
6.0	Muhimbi	07/06/2012	Updated for version 6.0
6.1	Muhimbi	26/09/2012	Updated for version 6.1

## Purpose and audience of document

This document describes the installation steps as well as general administrative topics related to the Muhimbi PDF Converter Services.

The intended audience is anyone involved in the installation and administration of this solution.

## Disclaimer

© Muhimbi. All rights reserved. No part of this document may be altered, reproduced or distributed in any form without the expressed written permission of Muhimbi.

This document was created strictly for information purposes. No guarantee, contractual specification or condition shall be derived from this document unless agreed to in writing. Muhimbi reserves the right to make changes in the products and services described in this document at any time without notice and this document does not represent a commitment on the part of Muhimbi in the future.

While Muhimbi uses reasonable efforts to ensure that the information and materials contained in this document are current and accurate, Muhimbi makes no representations or warranties as to the accuracy, reliability or completeness of the information, text, graphics, or other items contained in the document. Muhimbi expressly disclaims liability for any errors or omissions in the materials contained in the document and would welcome feedback as to any possible errors or inaccuracies contained herein.

Muhimbi shall not be liable for any special, indirect, incidental, or consequential damages, including without limitation, lost revenues or lost profits, which may result from the use of these materials. All offers are non-binding and without obligation unless agreed to in writing.

## Contents

1	Introduction	6
1.1	Relevant articles on the Muhimbi Blog	6
1.2	Prerequisites	8
1.3	High level solution architecture	8
2	Deployment	9
2.1	Installing the Windows Service	9
2.2	Installation Verification	10
2.3	Installing the License	11
2.4	Tuning the Document Conversion service	12
2.4.1	Authentication (Generic)	12
2.4.2	Authentication (from SharePoint)	13
2.4.3	Concurrency	13
2.4.4	Timeouts and File Size limitations	14
2.4.5	Logging	15
2.4.6	Adding custom converters / changing file extensions	15
2.4.7	Exception handling	16
2.4.8	InfoPath specific switches	16
2.4.9	HTML specific switches	17
2.4.10	Word processing (MS-Word) specific switches	18
2.4.11	Presentations (PowerPoint) specific switches	19
2.4.12	AutoCAD specific switches	19
2.4.13	Switches used for overriding settings	20
2.4.14	PDF Security Settings	21
2.5	Un-installation	22
2.6	Upgrading from a previous version	22
3	Troubleshooting & Other common tasks	23
3.1	Windows Event Log	23
3.2	Trace Log	23
3.3	Common issues & Errors	23
3.3.1	Error messages related to printer drivers or the printer spooler are logged	23
3.3.2	Problems parsing the WSDL	23
3.3.3	Documents using non standard fonts (e.g. Japanese) are not converted properly / The fonts in the destination document are not correct	24
3.3.4	Problems converting InfoPath forms without a shared XSN file	24
3.3.5	InfoPath forms using Ink controls fail to convert	24
3.3.6	Error 403 (Forbidden) when converting InfoPath forms	25
3.3.7	InfoPath files are converted using an old version of the XSN template	25
	Appendix - Installing converter dependencies	26
	Supported formats & their dependencies	26
	Using Office 2007	26
	Using Office 2010 / 2013	27
	Appendix - Using InfoPath with External Data Sources	28
	Details for InfoPath 2007	28
	Details for InfoPath 2010	30
	Digitally signing forms	30
	Using Muhimbi's 'AutoTrustForms' feature	30
	Appendix - Post processing PDF output to PDF/A	32
	Appendix - Unattended (un)installation	34



## PDF Converter Services - Installation & Administration Guide

Installation	34
Uninstallation	34
Upgrading	34
Appendix - Advanced Deployment Scenarios	35
Appendix - Creating Custom Converters	39
Appendix - Invoke 3 <sup>rd</sup> party Converters	44
Appendix - Licensing	46

## 1 Introduction

This document describes the installation steps as well as general administrative topics related to the Muhimbi PDF Converter Services.

The intended audience is anyone involved in the installation and administration of this solution. It is assumed that the audience has some familiarity with installing services on the Windows platforms and have been given the privileges to install and deploy solutions.

For more details about this product please see:

1. Product Information:  
<http://www.muhimbi.com/Products/PDF-Converter-Services.aspx>
2. Knowledge Base / Frequently Asked Questions:  
<http://support.muhimbi.com/>
3. Release Notes:  
<http://www.muhimbi.com/support/documentation/PDF-Converter-Services/Release-Notes.aspx>
4. User & Development Guide:  
<http://www.muhimbi.com/support/documentation/PDF-Converter-Services/User---Developer-Guide.aspx>
5. PDF Converter Service related content on the Muhimbi Blog:  
<http://blog.muhimbi.com/search/label/PDF%20Converter%20Services>

To keep on top of the latest news and releases, please subscribe to our blog or twitter feed at <http://www.muhimbi.com/contact.aspx>.

### 1.1 Relevant articles on the Muhimbi Blog

The Muhimbi Blog is updated frequently with new articles related to this product. At the time of writing the following posts are available.

- [Troubleshooting steps for the PDF Converter.](#)
- [Troubleshooting InfoPath to PDF Conversion / Document Converter Architecture.](#)
- [Convert Office files to PDF Format from .NET using a Web Service](#)
- [Invoking the PDF Converter Web Service from Visual Studio 2005 using vb.net](#)
- [Convert Office files to PDF Format from Java using a Web Service.](#)
- [Converting and merging multiple files using Web Services \(.NET\)](#)
- [Converting and merging multiple files using Web Services \(Java\)](#)
- [Splitting PDF Files using the PDF Converter Web Service and .NET / C#](#)
- [Using Windows Azure to convert documents to PDF format](#)
- [Adding custom Converters to Muhimbi's range of PDF Conversion products](#)
- [Using the awesome new watermarking features of the Muhimbi PDF Converter Services](#)
- [Using the PDF Watermarking features from Java based environments](#)

- [Converting InfoPath forms including all attachments to a single PDF file](#)
- [Controlling which views to export to PDF format in InfoPath](#)
- [Dealing with hyperlinks when converting InfoPath files to PDF format](#)
- [Convert InfoPath to MS-Word, Excel, XPS and PDF](#)
- [Convert between document types \(xls to xlsx, doc to docx, xls to doc\)](#)
- [Convert HTML pages to PDF format](#)
- [Converting AutoCAD \(DXF, DWG\) files to PDF](#)
- [Converting TIFF files to PDF](#)
- [Convert Outlook MSG files to PDF including all attachments](#)
- [PDF/A Support in the Muhimbi PDF Converter Services & SharePoint](#)
- [Converting PDF document to PDF/A1b using a Web Service](#)
- [Reduce PDF Converter Web Service message size using MTOM](#)

A number of articles written for SharePoint based environments are available as well.

- [Adding the PDF Converter to non standard Document and Form Libraries](#)
- [Tuning SharePoint's workflow engine](#)
- [Using the PDF Converter from a SharePoint Designer workflow](#)
- [Convert and merge multiple PDF files using a SharePoint Designer workflow](#)
- [Converting multiple SharePoint files to PDF Format using Nintex workflow](#)
- [Watermark PDFs using Nintex Workflow](#)
- [Secure PDFs using Nintex Workflow](#)
- [Convert and Merge PDFs using Nintex Workflow](#)
- [Convert HTML to PDF using Nintex Workflow](#)
- [Convert SharePoint documents to PDF using K2 workflows](#)
- [SharePoint Designer workflows trigger before properties are set](#)
- [Inserting SharePoint List data into a PDF document using a workflow](#)
- [Configure PDF Security from a SharePoint Designer Workflow](#)
- [Watermarking features of the Muhimbi PDF Converter for SharePoint](#)
- [Applying user specific watermarks when a PDF document is opened](#)
- [Merging dynamic data into watermarks using the PDF Converter for SharePoint](#)
- [Use SharePoint Workflows to inject JavaScript into PDFs and print the 'open date'](#)
- [Using the PDF Converter for SharePoint from your own code](#)
- [Automatically convert files to PDF using an e-mail enabled Document Library](#)
- [Batch print InfoPath Forms using the PDF Converter for SharePoint](#)
- [Using SharePoint Forms Services to convert InfoPath forms to PDF format](#)
- [Convert SharePoint HTML pages to PDF format](#)
- [Converting SharePoint Lists to PDF format using a SharePoint Designer Workflow](#)
- [Convert and merge files to PDF using the SharePoint User Interface](#)
- [Automatically mirror / sync a SharePoint folder structure using the Workflow Power Pack](#)
- [Create Shortened \('TinyURL'\) links from your SharePoint Workflow](#)
- [Send rich emails with attachments from a SharePoint Designer Workflow](#)

## 1.2 Prerequisites

The solution has been designed to work on a wide number of platforms. The prerequisites are as follows:

Server O.S.	Windows Server 2003 (including R2) 32 / 64 bit Windows Server 2008 (including R2) 32 / 64 bit
Client O.S.	The product is OS agnostic
Supported Languages	Any language that supports Web Services including .NET and Java.
Office Version	Office 2007 (SP2) / 2010 / 2013 applications for the relevant converters.
System Memory	This depends on the size and complexity of the documents that are converted and the number of concurrent conversions taking place. We recommend a minimum of 1GB of total memory.
CPU	Any CPU that can comfortably run the selected Operating System will be suitable. We recommend one or more multi-core CPUs.
Disk Space	This Product requires 20MB of space

## 1.3 High level solution architecture

The *Muhimbi PDF Converter Services* is a highly optimised solution for programmatically converting, watermarking and securing documents created in typical MS-Office applications, as well as other formats such as HTML, MSG (email), AutoCAD and images, to PDF or XPS format using any Web Services based environment including Java and .NET.

The converter runs as a Windows Service that can be deployed to either a separate system / virtual machine or to the server hosting your own application.

Although the actual converter must be installed on a Windows based environment, it can be invoked from any platform that supports Web Service calls including Windows, Linux, Solaris, AIX and Mac OS X.

Conversions can be *scaled up* by running multiple conversions in parallel and *scaled out* using standard HTTP Load balancers. For details see *Appendix - Advanced Deployment Scenarios*.

To achieve optimal conversion quality, for some file formats the Conversion Service uses MS-Office's own libraries in the background to carry out the actual conversion. Muhimbi's software stack ensures that this happens in a reliable manner without taking up excessive system resources. Common and uncommon problems are detected and corrective action is taken automatically without requiring any attention from system administrators.



## 2 Deployment

*Unless your solution will be installed in a single server environment, it is worth reading [Appendix - Advanced Deployment Scenarios](#)*

### 2.1 Installing the Windows Service

Please carry out the following steps to install the service and all prerequisites:

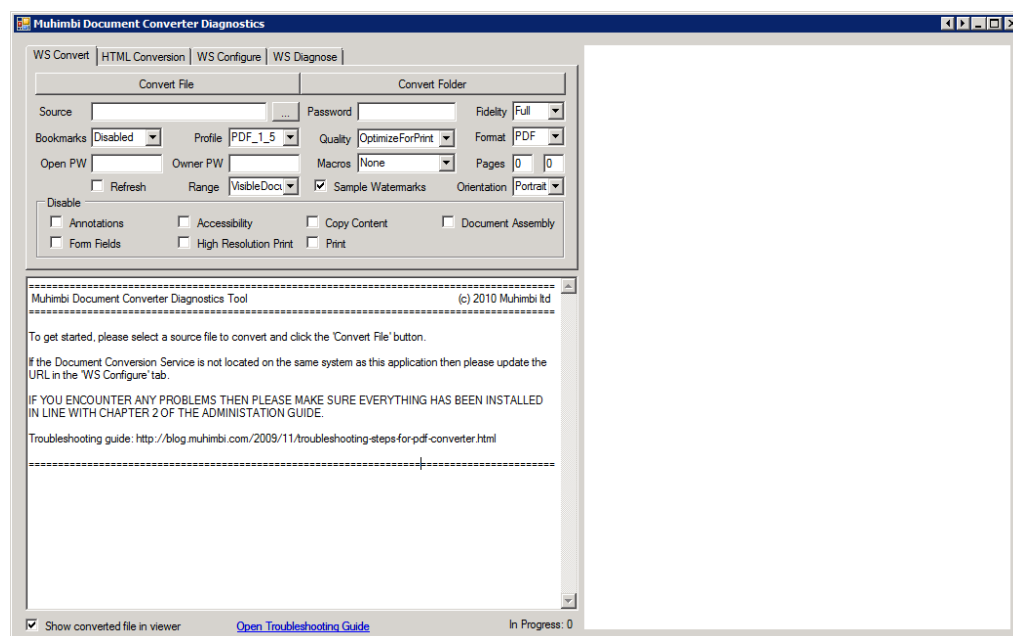
1. Download the software from the Muhimbi site and unzip it.
2. If a copy of the solution is already installed then please un-install it first. For details see [2.6 Upgrading from a previous version](#).
3. The service will need to run under a user account with the following privileges. When the software is installed for evaluation purposes then it is recommended to use the developer's or administrator's local account providing it is configured as follows:
  - a. Local Administrator.
  - b. *Log on as a Service* right.
  - c. If the software will be used to convert InfoPath Forms then the account will need to be able to access the XSN file, possibly located on a UNC path or in SharePoint, so make sure it has read access wherever the XSN file is located.
4. Install the Conversion Service by launching '*setup.exe*'.
5. Follow the instructions and enter the account defined in step 3 when prompted. For domain accounts please include the domain name. For local accounts use the '.' prefix, e.g. '.\Administrator').
6. Some of the converters have dependencies on external applications, see [Appendix - Installing converter dependencies](#). **Do not skip this step!**

Although the Out-Of-The-Box settings are suitable for most scenarios, you may want to change the settings as described in section [2.4 Tuning the Document Conversion service](#).

It is also possible to carry out a *quiet* or *unattended* install. For details see [Appendix - Unattended \(un\)installation](#)

## 2.2 Installation Verification

To verify the software and all prerequisites have been setup and configured correctly, execute the Diagnostic tool located in the Start Menu.



The Diagnostics Tool exposes the following functionality:

- **WS Convert:** Individual files as well as folders can be converted from this tab. Although the default settings are sufficient for most conversions, feel free to play around with the various settings.
- **HTML Conversion:** Convert a URL or HTML fragment to PDF format.
- **WS Configure:** Request configuration data detailing what converters are available and which file extensions they know how to process. The location of the Web Service, as used by the other tabs, can be configured on this tab as well.
- **WS Diagnose:** The button available on this screen invokes each of the individual converters using an internal test file to check if everything has been installed correctly.

Clicking the button on the *WS Diagnose* tab verifies if the installation has been carried out correctly.

Note that the *Diagnostics Tool* is not used to configure the Conversion Service. It is merely a test harness that can be used to test if the system has been installed correctly. The full source code of this tool is included and can be accessed from the Start Menu.

## 2.3 Installing the License

In order to run a fully licensed version of the software without any 'trial messages', you will need to copy the license file to the directory where the Conversion Service has been installed in. A shortcut to this folder (Open Installation Folder) can be found in the Windows Start Menu.

If you have purchased a license for the *PDF Converter Professional* as well then please make sure that license file is added using the same steps. Please note that a *PDF Converter Professional* license cannot be used unless a license for the *PDF Converter Service* is also in place.

After copying the license files restart the Conversion Service using the *Windows Services* MMC or the command line.

```
Net stop "Muhimbi Document Converter Service"
```

```
Net start "Muhimbi Document Converter Service"
```

## 2.4 Tuning the Document Conversion service

The settings for the Document Conversion Service can be changed by editing the *Muhimbi.DocumentConverter.Service.exe.config* file located in the directory the Conversion Service has been installed in<sup>1</sup>.

The various settings that can be changed are discussed below. Note that the Service must be restarted after making changes to the configuration file. Use the *Windows Services* MMC or the command line to do this:

```
Net stop "Muhimbi Document Converter Service"
```

```
Net start "Muhimbi Document Converter Service"
```

*Please note that some additional settings related to the post processing of PDF/A files can be found in Appendix - Post processing PDF output to PDF/A*

### 2.4.1 Authentication (Generic)

To make the initial installation as simple as possible, particularly for environments that access the Conversion Service from non Windows based platforms, anonymous access is enabled by default.

Although in general Production environments are shielded by a firewall, depending on your organisation you may want to enable an extra layer of authentication.

Authentication and Authorization are controlled by the following attributes and elements in the *Config* file:

- **ConversionClientsGroup:** The name of the Windows group that contains the accounts that are allowed to carry out conversions.
- **ConversionAdministratorsGroup:** The name of the Windows group that contains the accounts that can execute typical Administrative tasks such as running diagnostics.
- **Security mode:** Either use *TransportCredentialOnly* or *None*.
- **ClientCredentialType:** The type of credential used for client authentication. Either use *Windows* or *None*.

The Document Conversion Service uses Microsoft's Windows Communication Foundation (WCF) framework. For further details about configuring security have a look at Microsoft's MSDN site at <http://msdn.microsoft.com/en-us/library/ms731925.aspx>.

The following table contains a number of common scenarios:

	Conversion Clients Group	Conversion Administrators Group	Security mode	Client Credential Type
Anonymous	<leave empty>	<leave empty>	"None"	"None"
SharePoint	wss_wpg	wss_admin_wpg	TransportCredentialOnly	Windows

<sup>1</sup> The Start menu contains a shortcut to this folder.

### 2.4.2 Authentication (from SharePoint)

If you intend to use the Document Conversion Service from a SharePoint environment, then it is recommended to configure security as per the table in section 2.4.1.

This will restrict use to members of the standard SharePoint *wss\_wpg* and *wss\_admin\_wpg* groups. These groups, however, are local to the SharePoint machine, which may cause problems if the Document Conversion Service is installed on a separate system that does not have these local groups.

The solution is to either manually create these groups on the server hosting the Document Converter, and populate them with the same users as on the SharePoint servers, or to change the name of the groups in the config file.

The group names are defined in the following 2 configurations keys:

- **ConversionClientsGroup:** For SharePoint set this to *wss\_wpg*
- **ConversionAdministratorsGroup:** For SharePoint set this to *wss\_admin\_wpg*

If there is no need to restrict access to the back end of the Document Converter Service then you may want to consider changing the group names to '*NT AUTHORITY\authenticated users*'.

The *authenticated users* group allows any user with a valid login account to connect. Note that this is not the same as anonymous access.

Keep in mind that SharePoint connects to the Document Conversion service using the Web Application's *application pool* account, not the user's account.

### 2.4.3 Concurrency

The Document Converter allows multiple conversion requests to be processed simultaneously. The default settings are sufficient for most situations, but if you are running the service on a standalone server or if you expect the majority of your conversions to be of a single specific format then you may want to tune the concurrency settings.

The following settings can be changed in the config file.

- **serviceThrottling maxConcurrentCalls:** This setting represents the maximum number of concurrent requests that can be executed across all operations before new requests are queued. Be careful when lowering the default value when using the product to convert InfoPath forms with attachments or MSG files as this may result in a *deadlock* situation.

Please note that this number includes any requests for applying watermarks or Security on documents that are already in PDF format and don't require conversion.

- **Concurrency.MaximumInstances.WinWord:** The maximum number of concurrent MS-Word conversion requests before new requests are queued. This value defaults to 2.
- **Concurrency.MaximumInstances.Excel:** The maximum number of concurrent Excel conversion requests before new requests are queued. This value defaults to 2.

- **Concurrency.MaximumInstances.Visio:** The maximum number of concurrent Visio / Vector conversion requests before new requests are queued. This value defaults to 2.
- **Concurrency.MaximumInstances.CAD:** The maximum number of concurrent AutoCAD conversion requests before new requests are queued. This value defaults to 2.
- **Concurrency.MaximumInstances.TIFF:** The maximum number of concurrent TIFF conversion requests before new requests are queued. This value defaults to 2.
- **Concurrency.MaximumInstances.MSG:** The maximum number of concurrent MSG (email) conversion requests before new requests are queued. This value defaults to 2.
- **Concurrency.MaximumInstances.PowerPNT:** Do **not** change this value as PowerPoint does not allow concurrent requests.
- **Concurrency.MaximumInstances.MSPub:** The maximum number of concurrent Microsoft Publisher conversion requests before new requests are queued. This value defaults to 2.
- **Concurrency.MaximumInstances.InfoPath:** Changing this value only impacts the parallel processing of InfoPath attachments. InfoPath itself does not allow concurrent requests.
- **Concurrency.MaximumInstances.HTML:** The maximum number of concurrent HTML and Image conversion requests before new requests are queued. This value defaults to 2.

### 2.4.4 Timeouts and File Size limitations

To prevent users from sending overly complex documents to the conversion service and blocking access for other users for a long amount of time, it is possible to restrict the maximum time a conversion process is allowed to run. The following settings can be changed to deal with long running conversions:

- **Maximum run duration:** By default individual conversion requests are not allowed to run for more than 10 minutes. This should be sufficient for even the most complex documents. However, this value can be changed as follows:
  - Change the value of *ProcessMonitor.MAX\_RUN\_DURATION*.
  - Change the value of *receiveTimeout* to the same amount.
- **Maximum file size:** By default the maximum size of a file is 50MB. This value can be changed as follows:
  - *maxBufferSize*: Specify the new maximum file size in bytes.
  - *maxReceivedMessageSize*: Enter the same value here.
  - *maxArrayLength*: Enter the same value here.
  - *maxStringContentLength*: Enter the same value here.

### 2.4.5 Logging

The Document Conversion Service uses the industry standard *log4net* framework to write logging and trace data to a log file. Out-of-the-box information is logged to the *DocumentConverter.log* file stored in the directory the Document Conversion service has been installed in. A new file is created for each day and the default logging level is set to 'INFO'.

Warnings and Errors are also written to the Windows Event Log.

You may want to consider changing the following settings:

- **Log file location:** change the path of the log file name in the *appender* element to a location of your preference.
- **Log Level:** By default only 'INFO' and critical events are logged. To get a better view of what the service is doing, e.g. during a debug session, you may want to consider switching the <root> log level to DEBUG mode.

More information can be found at <http://logging.apache.org/log4net/index.html>.

### 2.4.6 Adding custom converters / changing file extensions

The Document Conversion Service makes it possible to add new converters as well as change the file extensions managed by each converter. This makes it possible to add new converters that are not shipped with the product as well as control which file types each converter deals with.

This information is stored in the *MuhimbiDocumentConverters* element in the service's config file, for example:

```
<add key="WordProcessing"
  description="Word Processing (e.g. MS-Word, RTF, TXT)"
  fidelity="Full"
  supportedExtensions="doc,docx,docm,rtf,txt,wps,xml,eml,odt"
  supportedOutputFormats="xps,pdf,doc,docx,rtf,txt,html,mht,xml,odt"
  type="Muhimbi.DocumentConverter.WebService.WordProcessorConverterFull
    Fidelity, Muhimbi.DocumentConverter.WebService, Version=1.0.1.1,
    Culture=neutral, PublicKeyToken=c9db4759c9eaaad12" />
```

Each converter has the following attributes:

- **key:** The name of the converter. Do not change this for existing converters unless absolutely needed as client side applications may store information against this key.
- **description:** An optional, human readable, description of the converter. This value may be used by client side application to display details about the converter.
- **fidelity:** The system has the notion of Full Fidelity and High Fidelity converters. When programming against the Web Services interface it is possible to select which fidelity to use.

This makes it possible to have 2 different converters that deal with the same file type. For example MS-Word can be used to convert complex documents (Full Fidelity) whereas a streamlined - high performance - third party component can be used to convert simple files at very high speed (High Fidelity).



- **supportedExtensions:** A comma separated list of file extensions that are recognised by the converter.
- **supportedOutputFormats:** A comma separated list of file types that the converter can generate.
- **type:** The full .NET type of the converter.

For details about how to create custom converters see *Appendix - Creating Custom Converters*.

### 2.4.7 Exception handling

By default any exceptions that occur in the Web Services are passed to the calling client application including a full stack trace. If this is not desired then set the *includeStackTraceInFaultReason* key in the config file to *false*.

As a result Exceptions will still be thrown, but the full stack trace will not be included.

### 2.4.8 InfoPath specific switches

To keep InfoPath installation relatively straight forward, the Document Conversion Server makes certain modifications to the files before conversion takes place. In some cases this is not desirable, for example if data from an external data source is used for display purposes, or if certain embedded code must run before conversion.

If additional control over the InfoPath conversion is required then consider modifying the following settings.

- Should InfoPath forms marked as *requiring Full Trust* be processed based on the parameters below (e.g. *StripDotNETCode*) or not ? Generally leave the default value of 'true' unless your XSN file is digitally signed  
`<add key="InfoPathConverterFullFidelity.ProcessFullTrustForms" value="true"/>`
- Remove all .net code from the form before conversion. Defaults to *True*.  
`<add key="InfoPathConverterFullFidelity.StripDotNETCode" value="true"/>`
- Remove all external data connections before conversion. Defaults to *True*.  
`<add key="InfoPathConverterFullFidelity.StripDataObjects" value="true"/>`
- Process any rulesets that may be present. Defaults to *True*.  
`<add key="InfoPathConverterFullFidelity.ProcessRuleSets" value="true"/>`
- Some forms with complex rules such as *get-SharePointServerRootUrl()* require the trust level to be restricted. When this value is set to true *StripDotNETCode* and *StripDataObjects* *\*MUST\** be set to 'true' as well.  
`<add key="InfoPathConverterFullFidelity.RestrictTrustLevel" value="false"/>`

Note that modifying these options makes InfoPath configuration more complex. Do not change these settings unless you have a good reason to do so. For details see *Appendix - Using InfoPath with External Data Sources*.



Although we are happy to assist, we cannot guarantee that the PDF Converter will operate correctly when the before mentioned settings have been modified. Having said that, many of our customers use the PDF Converter without any problems using custom settings.

By default any attachments found in an InfoPath form are converted to PDF and merged into a single PDF file. This behaviour can be disabled using the following configuration key.

```
<add key="InfoPathConverterFullFidelity.ConvertAttachments" value="true"/>
```

In order to convert InfoPath forms to PDF the converter retrieves the XSN file associated with the InfoPath XML file. By default it uses the Muhimbi Service's credentials when downloading this XSN file, but in some environments this may not be desirable. For those situations it is possible to override the credentials using the following configuration keys:

```
<add key="InfoPathConverterFullFidelity.XSNUserName" value=""/>
```

```
<add key="InfoPathConverterFullFidelity.XSNPassword" value=""/>
```

```
<add key="InfoPathConverterFullFidelity.XSNDomain" value=""/>
```

For details about the *InfoPathConverterFullFidelity.AutoTrustForms* setting see *Appendix - Using InfoPath with External Data Sources*

InfoPath internally caches files for each request. Over time this may use up a lot of space on the server's hard disk. By default this cache is cleared by the PDF Converter once an hour, but this duration can be changed using the *InfoPathConverterFullFidelity.XSNCacheClearInterval* setting.

#### 2.4.9 HTML specific switches

Due to the way certain websites (e.g. SharePoint) behave when JavaScript is disabled, a global override flag has been added to the config file to force JavaScript to be enabled when carrying out HTML based conversions.

This override can be disabled using the following configuration key.

```
<add key="HTMLConverterFullFidelity.ForceJavaScript" value="False"/>
```

When disabled, the value in the *ConversionSettings.MacroSecurityOption* web service property is used to control if JavaScript should be enabled or not.

It is also possible to adjust the default *Paper Size*, *Page Orientation*, *Margins* and *Scale Mode* for the HTML Converter. For full details see the *HTMLConverterFullFidelity* keys in the service's config file.

These settings can be overridden on a request-by-request basis using the *ConverterSpecificSettings* property on the web services interface.

When experiencing problems converting SharePoint 2010 pages then make sure *HTMLConverterFullFidelity.ClearBrowserCache* is set to *True*.

### 2.4.10 Word processing (MS-Word) specific switches

The following settings are specific to the Word processing based converter and can be specified either globally for all requests using the service's config file or on a *request-by-request* basis using the *ConverterSpecificSettings* property on the web services interface.

- **RevisionsAndCommentsDisplayMode:** Specify the default value for how to view the proposed changes to the document.
  - **FinalShowingMarkup:** Convert the document with all proposed changes highlighted.
  - **Final:** Convert the document with all proposed changes included.
  - **OriginalShowingMarkup:** Convert the original document with all proposed changes highlighted.
  - **Original:** Convert the document before any changes were made.

```
<add key="WordProcessorConverterFullFidelity.RevisionsAndCommentsDisplayMode" value="Final"/>
```
- **RevisionsAndCommentsMarkupMode:** Specify the default value for how to visualise revisions to the document. You can show revisions as balloons in the margins of the document or show them directly within the document itself.
  - **InLine:** Show all revisions Inline
  - **Balloon:** Show all revisions in balloons
  - **Mixed:** Show only comments and formatting in balloons

```
<add key="WordProcessorConverterFullFidelity.RevisionsAndCommentsMarkupMode" value="InLine"/>
```
- **ProcessDocumentTemplate:** Specify if the MS-Word template will need to be stripped out for DOCX files. Leave the default setting (true) unless formatting problems occur.
- **ProcessDOCFiles:** Pre-convert DOC files to DOCX to allow processing using the above mentioned *ProcessDocumentTemplate* setting. Only enable this option when experiencing problems with Document Information Panels. Note that there may be some side effects so talk to support@muhimbi.com before changing this setting.

### 2.4.11 Presentations (PowerPoint) specific switches

The following settings are specific to the Presentations based converter and can be specified either globally for all requests using the service's config file or on a *request-by-request* basis using the *ConverterSpecificSettings* property on the web services interface.

- **PrintOutputType:** Specify the default value for the part of the presentation to print. Supported values are: *Slides*, *NotesPages*, *Outline*, *OneSlideHandouts*, *TwoSlideHandouts*, *ThreeSlideHandouts*, *FourSlideHandouts*, *SixSlideHandouts*, *NineSlideHandouts*.

```
<add key="PresentationConverterFullFidelity.PrintOutputType" value="Slides"/>
```

- **FrameSlides:** Specify the default value for the border around the slide. Accepts *True* or *False*.

```
<add key="PresentationConverterFullFidelity.FrameSlides" value="True"/>
```

### 2.4.12 AutoCAD specific switches

The AutoCAD converter is highly configurable and allows the following settings to be specified either globally for all requests using the service's config file or on a *request-by-request* basis using the *ConverterSpecificSettings* property on the web services interface

For full details search the service's config file for all keys starting with *CadConverterFullFidelity*.

- **PaperSize:** Specify the paper size of the generated PDF file, either using standardised paper size names such as *A4* or *Letter*, or using custom dimensions in inches or millimetres.
- **PageMargin:** Determine the margin around the page, either as a single uniform value or using a different value for *Left*, *Top*, *Right* and *Bottom*.
- **PaperOrientation:** Automatically detect the drawing's orientation or force it to *Portrait* or *Landscape*.
- **BackgroundColor:** Some AutoCAD drawings have been created using a non-white background colour, in which case you may want to force the background colour in the PDF file using this setting.
- **ForegroundColor:** The PDF Converter supports a number of automatic re-colouring options to make sure that foreground colours don't conflict with background colours or to convert all colours to grey scale. For details see the in-line documentation in the service's config file.
- **Content:** By default all valid layouts found in a DXF or DWG file are converted to PDF. However, you may want to override this setting and only convert specific layouts. This can be particularly useful when converting 3D AutoCAD files.
- **EmptyLayoutDetectionMode:** Specifies how the conversion handles empty or nearly empty layouts.
- **LayoutSortOrder:** Specify the sort order for layout names.

### MSG (email) specific switches

The MSG converter allows the following settings to be specified either globally for all requests, using the service's config file, or on a *request-by-request* basis using the *ConverterSpecificSettings* property on the web services interface.

For full details search the service's config file for all keys starting with *MSGConverterFullFidelity*.

- **PaperSize:** Specify the paper size of the generated PDF file, either using standardised paper size names such as *A4* or *Letter*, or using custom dimensions in inches or millimetres.
- **ConvertAttachments:** This config value determines if email attachments will be converted (and merged to the main email), or not.

### 2.4.13 Switches used for overriding settings

The following settings can be changed to allow any client side settings to be globally overridden. *Unless you have a specific reason to change them, leave these settings alone.*

- **Merging.ForceBreakOnError:** Override the *BreakOnError* value during merge. Leave empty to use the setting specified in the web service call (*MergeSettings*). Specify 'True' to fail the entire operation when an error occurs. Use 'False' to continue and skip any problematic files.
- **Merging.StripDigitalSignatures:** When merging PDF files (or InfoPath / MSG attachments) that contain digital signatures then please set this value to 'true' to remove all digital signatures. Signed documents cannot be merged.
- **OpenOptions.ForceAllowMacros:** Override the *AllowMacros* value during conversion. Leave empty to use the setting specified in the web service call (in *OpenOptions*). Accepted values are members of the *Muhimbi.DocumentConverter.WebService.Data.MacroSecurityOption* enumeration or an empty string.
  - None - Don't allow any macros to run
  - SignedOnly - Only allow macros to run with a valid digital signature
  - All - Allow all macros, regardless of their origin and digital signature. This is not recommended as it may compromise the security of the server.
- **OpenOptions.ForceRefreshContent:** Override the *RefreshContent* value during conversion. Leave empty to use the setting specified in the web service call (in *OpenOptions*). Accepted values are 'true', 'false' or empty string.
- **ConversionSettings.ForceQuality:** Override the *Quality* value during conversion. Leave empty to use the setting specified in the web service call (in *ConversionSettings*). Accepted values are members of the *Muhimbi.DocumentConverter.WebService.Data.ConversionQuality* enumeration or an empty string.
  - OptimizeForPrint - Optimise the file size and resolution for print purposes.
  - OptimizeForOnScreen - Optimise the file size and resolution for use on a computer screen

- **ConversionSettings.ForceRange:** Override the *Range* value during conversion. Leave empty to use the setting specified in the web service call (in *ConversionSettings*). Accepted values are members of the *Muhimbi.DocumentConverter.WebService.Data.ConversionRange* enumeration or an empty string.
  - VisibleDocuments - Skips, in case of Excel and PowerPoint, any hidden tabs or slides.
  - AllDocuments - Export all tabs or slides in a workspace.
  - ActiveDocuments - Exports, in case of Excel, the selected tabs.
- **ConversionSettings.ForceGenerateBookmarks:** Override the *GenerateBookmarks* value during conversion. Leave empty to use the setting specified in the web service call (in *ConversionSettings*). Accepted values are members of the *Muhimbi.DocumentConverter.WebService.Data.BookmarkGenerationOption* enumeration or an empty string.
  - Disabled - Don't generate any bookmarks.
  - Automatic - Based on headings, if applicable.
  - Custom - Based on bookmarks defined in the document, e.g. MS-Word Bookmarks, if applicable.
- **ConversionSettings.ForcePDFProfile:** Override the *PDFProfile* value during conversion. Leave empty to use the setting specified in the web service call (in *ConversionSettings*). Accepted values are members of the *Muhimbi.DocumentConverter.WebService.Data.PDFProfile* enumeration or an empty string.
  - PDF\_1\_5 - Use PDF Version 1.5
  - PDF\_A1B - Use the PDF/A standard for long term archiving

### 2.4.14 PDF Security Settings

The default PDF Security and Encryption settings have been designed to be compatible with the widest possible range of PDF Readers. However, this means that security has been 'dumbed down' a bit. If all your users are on relatively recent versions of PDF Readers then you may want to tighten up the security settings.

- **PDF.EncryptionKeySize:** Set the security key size for encrypting PDF files. Possible values are 40, 128 and 256. Please note that 256 Bit is not supported on versions of Acrobat older than version 9 and many other PDF readers.
- **PDF.EncryptionAlgorithm:** Set the security algorithm, either *RC4* or *AES*. Although cryptographically stronger, be careful when using *AES* as only Acrobat Reader 7 or newer can open *AES* encrypted files.
- **PostProcessSecuredFiles:** Define whether PDF files that are already secured will be post processed by the system. Post processing can include watermarking and applying security settings.

## 2.5 Un-installation

The Conversion Service can be uninstalled using the standard Windows *Programs and Features* control panel. (also known as *Add / Remove Programs*).

## 2.6 Upgrading from a previous version

Upgrading the software from a previous version is simple and straight forward. However, there are some things to take into account.

The Conversion Service uses a standard *.config* file to control the default behaviour of the product (see 2.4 *Tuning the Document Conversion service*). Please note that any changes to this file are not maintained during an upgrade. You will need to manually re-apply any changes that may have been made since the previous installation.

The upgrade process is as follows:

1. Uninstall the software as described in 2.5 *Un-installation*.
2. Install the new version as described in 2.1 *Installing the Windows Service*.
3. Make any of the post installation changes described in the introduction of this section.

## 3 Troubleshooting & Other common tasks

This section provides some guidance about troubleshooting problems. If you still have questions after reading this chapter then please check out the links in chapter 1 *Introduction*.

### 3.1 Windows Event Log

The following entries may be written to the event log:

1. **Warnings:** If you are running an evaluation copy of the software, or your license has expired then this is reported as a warning message in the Application Event Log.
2. **Errors:** All errors are written to the event log. The main cause of errors are corrupt documents.

Note that all event entries written by Muhimbi's products use Event ID 41734.

### 3.2 Trace Log

The Document Converter Service maintains a detailed trace log named *DocumentConverter.log* in the directory the service has been installed in.

For details on how to change the log settings see section 2.4.5 *Logging*.

### 3.3 Common issues & Errors

#### 3.3.1 Error messages related to printer drivers or the printer spooler are logged

The Excel and InfoPath converters require the *Printer Spooler* service to be started. The installer used by the Document Conversion Service attempts to start the Printer Spooler service automatically, but if the service is subsequently disabled then you may receive an error.

These converters also require at least one printer driver to be installed. Although in general it doesn't matter which driver is installed, some drivers such as VMWare's *Virtual Printer* or the *OneNote printer* will cause problems. Windows Server 2003 R2 and Windows Server 2008 automatically install the XPS driver, it is recommended to make this the default printer.

#### 3.3.2 Problems parsing the WSDL

By default the MDCS uses *localhost* as the base address. Most web service client libraries deal with this correctly, however if the service is exposed using a different machine name then you may need to update the *base address*.

In order to change this, modify the *baseAddress* attribute in the configuration file and restart the service.



### 3.3.3 Documents using non standard fonts (e.g. Japanese) are not converted properly / The fonts in the destination document are not correct

If the fonts in the converted document are not displayed properly then please make sure that the correct fonts are installed on the server hosting the Document Conversion Service.

For example, if Japanese characters are not displayed properly then make sure the Japanese language pack for Windows Server is installed.

If a font cannot be found then the system will attempt to find an alternative, but similar, font. If an alternative cannot be found then the system will use the Times New Roman font.

When using the optional *PDF/A post processing facility* then all used fonts are automatically embedded in the PDF document. If fonts are deployed to the server after Ghostscript was installed then please execute the following command from the Ghostscript bin directory.

```
gswin32c -q -dBATCH -sFONTDIR=%windir%/fonts -sCIDFMAP=../lib/cidfmap ../lib/mkcidfm.ps
```

Please replace *gswin32c* with *gswin64c* when using the 64 bit version of Ghostscript. For more details about how the optional Ghostscript installation is used for PDF/A post processing see *Appendix - Post processing PDF output to PDF/A*.

AutoCAD fonts can be placed in the *SHXFonts* folder under the Conversion Service's installation folder.

### 3.3.4 Problems converting InfoPath forms without a shared XSN file

If you intend to convert InfoPath forms that don't rely on a shared XSN template stored on a file share or in SharePoint then the XSN file must be registered on the Document Conversion Server using RegForm.exe.

The RegForm application can be found in C:\Program Files\Microsoft Office\Office12. For details see the second half of the following article:

<http://msdn.microsoft.com/en-us/library/bb251025.aspx>

### 3.3.5 InfoPath forms using Ink controls fail to convert

When using Ink Controls, for example to capture signatures on InfoPath forms, then please make sure the Ink controls are installed on the machine(s) that run the Muhimbi Conversion Service. For example, in Windows Server 2008 these controls are installed as part of the "Ink & Handwriting features", part of the "Desktop Experience" Windows Feature.



### 3.3.6 Error 403 (Forbidden) when converting InfoPath forms

When InfoPath conversion fails and the error message in the Windows Application Event Log refers to “*The remote server returned an error: (403) Forbidden*” then this may be caused by the authentication type specified on the Web Application. If, for example, Forms Based Authentication is enabled, but NTLM authentication is not configured on the same Web Application then the Muhimbi Conversion Service will not be able to download the XSN file.

The problem can be solved by adding NTLM authentication to the Web Application.

### 3.3.7 InfoPath files are converted using an old version of the XSN template

Although rare, InfoPath sometimes re-uses an old XSN file when carrying out a conversion. Running the typical `InfoPath /cache ClearAll` command will not work as XSN files are stored in a different location when running as a service.

Please clear all *FormCache* folders manually from the following location:

#### Windows Server 2008 and newer

64bit: %windir%\SysWOW64\config\systemprofile\AppData\Local\Microsoft\InfoPath

32bit: %windir%\System32\config\systemprofile\AppData\Local\Microsoft\InfoPath

#### Windows Server 2003

C:\Documents and Settings\Default User\Local Settings\Application Data\Microsoft\InfoPath

## Appendix - Installing converter dependencies

To ensure the quality of the converted documents is as close to perfect as possible, some converters have a dependency on external applications such as MS-Office. These dependencies are only needed for conversions purposes, applying watermarks and PDF security have no dependencies.

When using our optional PDF/A post processing facility see *Appendix - Post processing PDF output to PDF/A* for some additional dependencies.

**These external applications must be installed on the system(s) that run the Muhimbi Document Conversion Service.** There is no need to install these dependencies on your Web Front End Servers unless these are also used to run the Conversion Service.

## Supported formats & their dependencies

Converter	Supported file types	Dependency
HTML & Web pages	html, htm, mht and any url that returns HTML such as .aspx or .jsp.	Internet Explorer
Image formats	gif, png, jpg, bmp, tif, tiff	-
AutoCAD formats	dwg, dxf	-
InfoPath forms	xml, infopathxml	InfoPath 2007/10/13
Word Processing	doc, docx, docm, rtf, txt, wps, xml, odt, ott, wpd	Word 2007/10/13
Emails	msg, eml	Word 2007/10/13, Internet Explorer
Spreadsheets	xls, xlsx, xlsx, xlsm, xlsb, xml, csv, dif, ods, ots	Excel 2007/10/13
Presentations	ppt, pptx, pptm, xml, odp, otp, pps, ppsx, ppsm	PowerPoint 2007/10/13
Publisher	pub	Publisher 2007/10/13
Vector formats	vsd, vdx, vdw, svg, svgz,	Visio 2007/10/13

## Using Office 2007

If needed, install the relevant MS-Office applications defined in the matrix above. **It is a requirement that at least Office 2007 Service Pack 2 is installed.**

Once installed, log-in to the server desktop (local or via remote desktop) using the account the *Muhimbi Document Conversion Service* will run under and launch each of the relevant Office applications, carry out Microsoft's activation process if needed, and close the application.

There is no need to reinstall / reactivate MS-Office when upgrading to a new version of the Muhimbi Converter as long as the service account remains the same.

### Using Office 2010 / 2013

If the Muhimbi PDF Converter will be used in combination with Office 2010 /13 then please follow the instructions set out below.

1. If needed, remove older versions of Office from the conversion server as environments with mixed Office versions on the same system are not supported by Muhimbi, even though it may work.
2. Install either the 32 or 64 bit version of Office 2010/13. Do not use any trial, un-activated or *click-to-run* version of Office.
3. If not already done so, log-in to the server desktop (local or via remote desktop) using the account the Document Conversion Service will run under and launch each of the relevant Office applications. Carry out Microsoft's activation process if needed, and close the application again.

There is no need to reinstall / reactivate MS-Office when upgrading to a new version of the Muhimbi Converter as long as the service account remains the same.

## Appendix - Using InfoPath with External Data Sources

*Please note that this appendix only applies to environments that have 'external data sources' enabled. Enabling data sources is a manual action, default deployments of the Muhimbi PDF Converter never ship with this option enabled. For more details see 2.4.8 'InfoPath specific switches'*

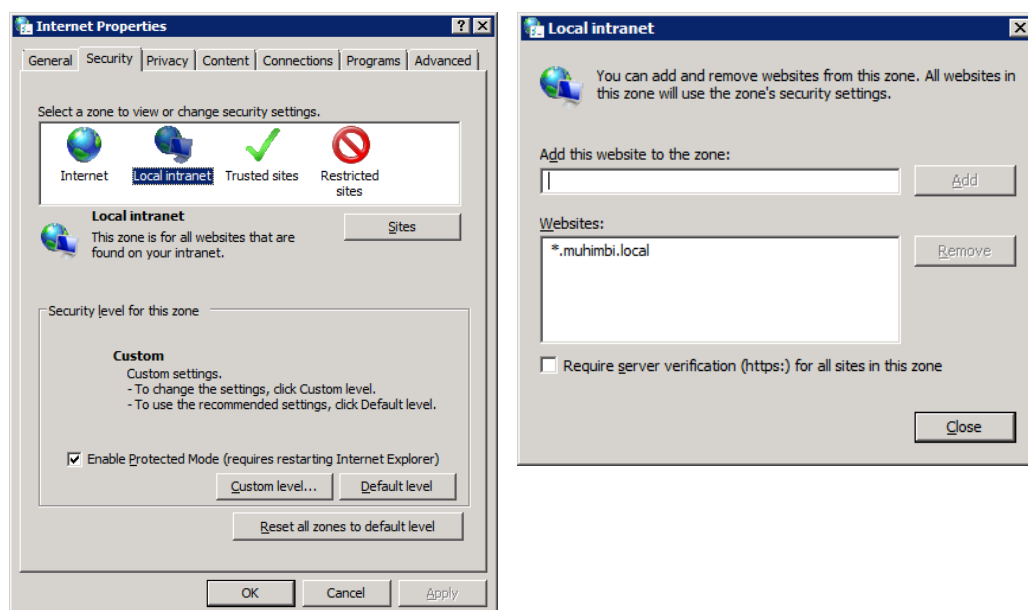
*Although we are happy to assist, we do not officially support conversions that run with 'StripDotNETCode', 'StripDataObjects' or 'ProcessRuleSets' set to 'false'. Having said that, many of our customers use the PDF Converter without any problems using these custom settings.*

### Details for InfoPath 2007

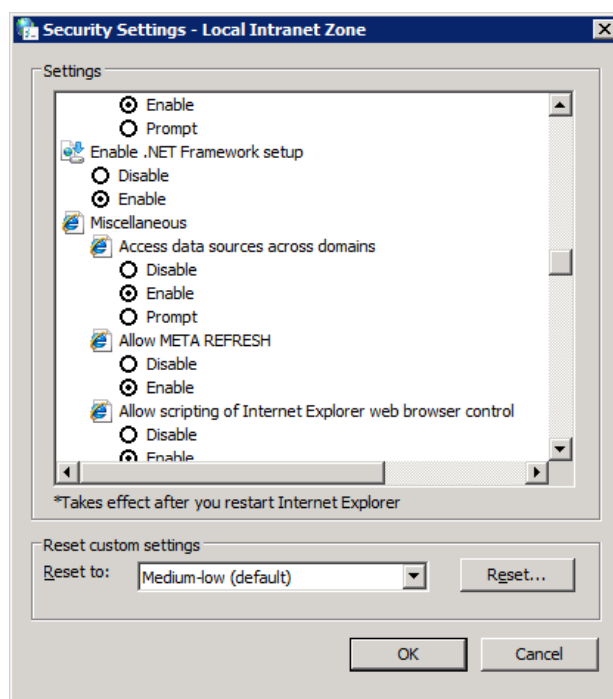
When an InfoPath document containing external connections, e.g. a dropdown list with the contents of a SharePoint list, fails to convert then this may be caused by the location of the XSN file not being trusted or the *access data sources across domains* setting not being enabled for the trusted site.

Ideally this configuration change should be made by a Domain Administrator using a group policy. However, the change can be made manually as well using the steps outlined below:

1. Log in using the account the *Muhimbi Document Converter Service* is running under.
2. Open *Internet Options* either from Internet Explorer or the Control Panel.
3. Verify the site that hosts the XSN file is recognised as a *Local Intranet* site by selecting *Local Intranet*, clicking the *Sites* button followed by the *Advanced* button. You may need to uninstall / disable *Internet Explorer Enhanced Security* on your server in order for this to work.

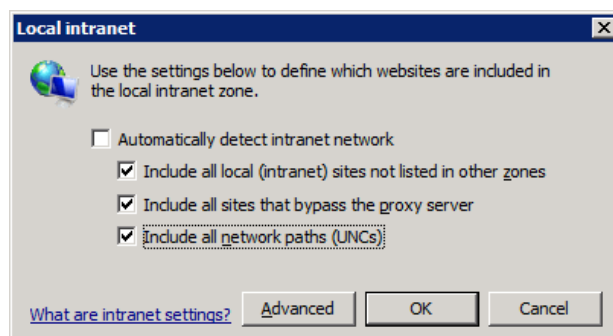


4. With *Local Intranet* selected, click the *Custom Level* button and verify that *Access data sources across domains* is set to *Enable*.



5. In the same screen navigate to the *User Authentication* section and select *Automatic Logon with current user name and password*.

If the previous steps fail to resolve the problem then make sure automatic detection of the Intranet network is disabled and the individual options (as per the screenshot below) are enabled.



This screen can be opened from *Internet Options / Security / Local Intranet / Sites*. Note that in Internet Explorer 6 the *Automatically detect intranet network* option is not present, but the individual options are.

If conversions of InfoPath documents don't work consistently then flush the local Form Template Cache by logging in as the service account and issuing the following command from Windows' Start / Run menu:

```
Infopath /cache clearall
```

### Details for InfoPath 2010

In InfoPath 2010 Microsoft has made changes to the way forms are trusted. As a result the instructions for InfoPath 2010 are different from those for InfoPath 2007.

#### Digitally signing forms

The most reliable way to convert forms *as-is*, with full support for Data Connections, Rule Sets and custom code, is to digitally sign the XSN file.

For details about how to do this see the following resources.

- <http://www.thorprojects.com/blog/archive/2007/06/30/domain-certificate-authority-signing-infopath-2007-forms.aspx>
- <http://msdn.microsoft.com/en-us/library/ee526348.aspx>
- <http://msdn.microsoft.com/en-us/library/ee526349.aspx>
- <http://msdn.microsoft.com/en-us/library/aa946782.aspx>

Please make sure to use a digital certificate that uses an *authority* that is trusted by the server that runs the Muhimbi Conversion Service. Temporary test certificates created on a development machine will not work when used on other machines.

When using digitally signed forms AND you wish to access external data connections and / or custom code during conversion then please set the *InfoPathConverterFullFidelity.ProcessFullTrustForms* key in the Muhimbi Service's config file to *false*.

#### Using Muhimbi's 'AutoTrustForms' feature

Starting with version 5.1 of the Muhimbi PDF Converter it is possible to convert InfoPath forms that use either the *Domain* or *Automatic* trust levels, with full support for External Data Sources, Rule Sets and running custom code.

To prevent compatibility problems with any previous versions of the software this feature is disabled by default. In order to enable it set the following value to *true* in the Muhimbi Service's config file and restart the service.

```
<add key="InfoPathConverterFullFidelity.AutoTrustForms" value="false"/>
```

Just enabling this feature by itself will achieve nothing, the idea is to disable at least one of the following options as well:

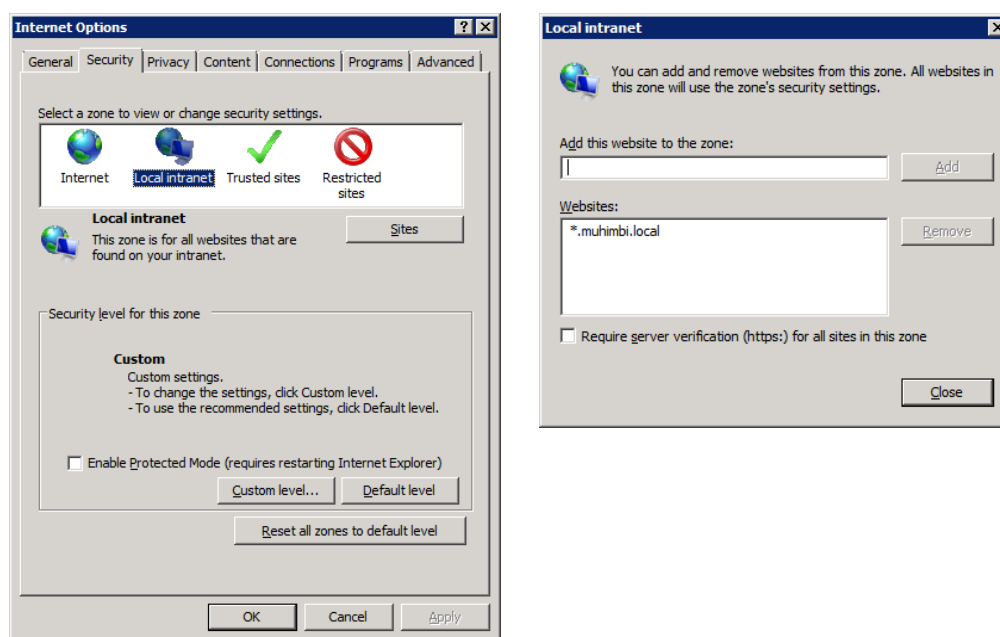
```
<add key="InfoPathConverterFullFidelity.StripDotNETCode" value="true"/>
<add key="InfoPathConverterFullFidelity.StripDataObjects" value="true"/>
<add key="InfoPathConverterFullFidelity.ProcessRuleSets" value="true"/>
```

For more details about these settings as well as editing the service's config file see 2.4.8 *InfoPath specific switches*.

In order to use this functionality file sharing must be enabled on the server that runs the Muhimbi Service and the default *Administrative Drive Shares* (C\$, D\$ etc) must be available as well.

When the *AutoTrustForms* facility is used to connect to external systems, e.g. a web service on your SharePoint server, then please make sure the server name is recognised as a *local intranet* system by adding the server name / entire domain to the list of *local intranet* sites. This can be done either manually or, even better, using a group policy. The manual steps are as follows:

1. Log in to the desktop using the account the Muhimbi Conversion Service runs under.
2. Start Internet Explorer and navigate to Tools / Internet Options / Security.
3. Select 'Local Intranet' and click the 'Default Level' button.
4. Click the 'Sites' button followed by 'Advanced'.
5. Add the server that is being connected to to the list.



Similar to the *Details for InfoPath 2007* section above, click the *Custom Level* button and make sure that *User Authentication* is set to *Automatic Logon with current user name and password* and that *Access data sources across domains* is set to *Enable*.



## Appendix - Post processing PDF output to PDF/A

Muhimbi's range of PDF Conversion products has provided some level of PDF/A support since the very beginning. We knew this was going to be important so we even built it into our web service's object model. However, up until version 5.1, any request to output the file in PDF/A format was passed on directly to the underlying converter, which may, or may not, do a good job of sticking to the PDF/A standard (more on this subject below).

This all changes with version 5.2 of our software, which allows any PDF file to be post processed and converted to PDF/A-1b. For more background information as well as additional details on this subject see the blog post at <http://blog.muhimbi.com/2011/09/pdfa-support-in-muhimbi-pdf-converter.html>.

### Configuring the Muhimbi PDF Converter to use PDF/A

The Muhimbi PDF Converter relies on 3rd party software to carry out the PDF/A post processing step. Fortunately this software is free to download and install for both individuals and organisations. The actual use of the software is less than trivial, but our software takes care of all the complexities.

Installation of this software is optional and only required if you intend to carry out any PDF/A post processing. The steps are as follows:

1. If not already installed, install the Muhimbi PDF Converter for SharePoint version 5.2 or newer.
2. Download the latest *GPL Release* [from the Ghostscript website](#). Depending on your hardware and operating system you will need to download either the 32 or 64 bit version. Muhimbi has tested the PDF/A post processing with version 9.04-9.06, but newer versions should work fine as well.
3. Install Ghostscript in a location of your choice on every server that runs the Muhimbi Conversion Service. If you accept the default location, or the default location on a different drive, then the Muhimbi PDF Converter will automatically detect Ghostscript's file path.

Please note that you need the [PDF Converter Professional](#) add-on license in addition to a valid *PDF Converter for SharePoint* or *PDF Converter Services* License in order to use this functionality.

Once Ghostscript has been installed you are ready to go, providing you access our Web Services based interface from your own software and the *PDFA.PostProcessing* configuration value discussed below is set appropriately (details for use with SharePoint can be found further below). Just set the web service's *ConversionSettings.PDProfile* property to *PDF\_A1B* and any converter that doesn't natively support PDF/A output will automatically send the generated PDF file to the post processor. However, depending on your exact needs you may want to update a number of settings in the Muhimbi Conversion Service's config file.

1. Open '*Muhimbi.DocumentConverter.Service.exe.config*' in your favourite text editor. The file is stored in the same directory where the Muhimbi Conversion Service has been installed in. You can find a shortcut to this folder in the '*Muhimbi Document Converter*' group in your start menu.



2. Change the following settings if needed:
  - **Ghostscript.Path:** Leave the path empty to auto detect the location. When manually specifying the path include the executable as well, e.g. "E:\Program Files\gs\gs9.04\bin\gswin64c.exe"
  - **PDFA.PostProcessing:** Not all converters are able to provide native PDF/A output. Use this setting to post process any generated PDF file. Valid values are 'All' (Post Process files generated by all converters, including the ones that are supposed to already support PDF/A), 'WhenNeeded' (Post process files for only those converters that do not support native PDF/A output) or 'None' (Do not post process files generated by any converters. This is the default option). Please note that these values will only be used if the output format is set to *PDF\_A1B*, either in the web service call or via the global '*ConversionSettings.ForcePDFProfile*' config value.
  - **PDFA.RasterizeTransparentContent:** Define how transparent content is dealt with during conversion to PDF/A. The default setting (False) removes all transparency. If you wish to retain transparent objects then set this value to *True*, which will result in pages being rasterized resulting in considerably larger and slower PDF files.
  - **ConversionSettings.ForcePDFProfile:** Override *ConversionSettings.PDFProfile* value during conversion. Leave empty to use the setting specified in the web service call. Accepted values are members of the *Muhimbi.DocumentConverter.WebService.Data.PDFProfile* enumeration or an empty string. For example: 'PDF\_1\_5' (Use PDF Version 1.5) or 'PDF\_A1B' (Use the PDF/A standard for long term archiving).
3. Restart the *Muhimbi Conversion Service* from the *Windows Services Management Console*.

If you don't directly interact with Muhimbi's Web Services interface, but rather use the SharePoint Front End functionality that comes with the PDF Converter for SharePoint, e.g. workflow activities or manual PDF Conversion, then you **MUST** set the *ForcePDFProfile* configuration value to *PDF\_A1B*. Please note that that this is a global switch that forces all functionality provided by our product to output in PDF/A. This may have side effects when applying PDF Security. In the future we will provide more granular support for those customers who rely on the SharePoint facilities that come with our products.

Sample code to convert PDF Documents to PDF/A using our web services interface [can be found here](#). If you don't wish to use Ghostscript, or if your organisation already uses a different PDF to PDF/A converter that you wish to integrate with, then please contact us on [support@muhimbi.com](mailto:support@muhimbi.com).

## Appendix - Unattended (un)installation

Many organisations carry out software deployments via a Software Distribution Service such as *Microsoft Systems Management Server*. This allows a product to be rolled out to one or more servers without user intervention. Muhimbi's PDF Converter can be installed in this fashion as described below.

### Installation

The Conversion Service installer follows the standard convention for deploying services in unattended mode. The syntax is as follows

```
msiexec /i setup.msi /quiet username="" password="" targetdir=""
```

The parameters are:

- **Username:** The user name the service will run under, including the domain name, e.g. '*muhimbi\conversion.service*'
- **Password:** The password for the account the service runs under.
- **Targetdir:** An optional parameter to specify the installation path. Remove the parameter to target the default installation path.

### Uninstallation

To uninstall the Conversion Service schedule the following command:

```
msiexec /x{C70C6D5A-4DBF-43FD-9EB8-41F352E948CC} /quiet
```

### Upgrading

The Conversion Service does not support in-place upgrades. When upgrading the software schedule the *uninstall* command followed by the *install* command.

For more details about upgrading see section 2.6 *Upgrading from a previous version*.

## Appendix - Advanced Deployment Scenarios

Muhimbi's range of server based PDF Conversion products have been developed with [performance, scalability and reliability](#) in mind. As a result the software scales from the most humble 'everything on the same server' environments to environments that deal with millions of conversions a day across a farm of servers. This appendix discusses the most common deployment scenarios.

Whenever this appendix mentions 'Server' it doesn't matter if this is a physical or virtualised server. The software does not differentiate between the two and will work fine on either type.

The latest version of this appendix, including user feedback, can be found on the Muhimbi blog at <http://goo.gl/qfMDk>.

### Introduction – Architecture

Both the PDF Converter for SharePoint and the PDF Converter Service ship with the same central conversion engine. This engine, *The Muhimbi Conversion Service*, is responsible for carrying out all the work including conversion of files, watermarking, merging, splitting and security activities. Although in case of the PDF Converter for SharePoint the front end is quite comprehensive, all it really does is prepare requests for the Conversion Service and receive responses containing new or modified documents.

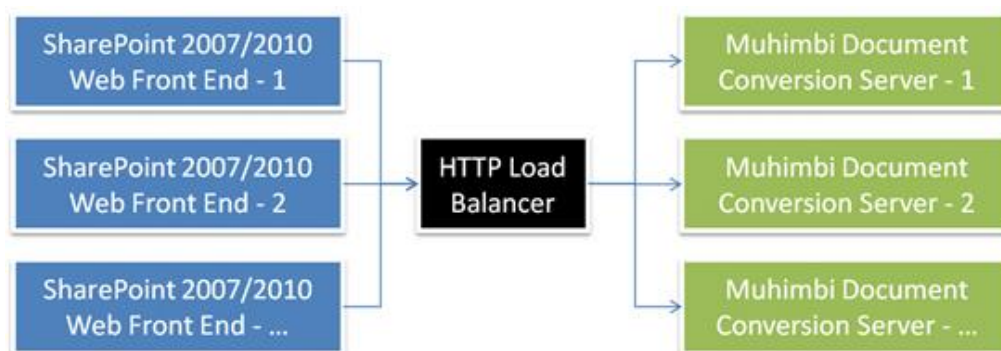
The Muhimbi Conversion Service is a standard Windows Service that starts automatically when Windows boots up and requires no user interaction or anyone to be logged in to the server console.

This Windows Service contains a WCF based Web Service that exposes functionality to any Web Services capable environment including *Java, C#, VB.NET, Documentum, SAP, Ruby, PHP* etc. Typically when administrators think about web services they assume that they need to host this inside a web server such as IIS or Apache. Although that may be true for many web services, Muhimbi's PDF Conversion software runs inside a *self-hosted* WCF service that does not have any external dependencies on third party web servers.

Basing the conversion service on WCF results in a lot of benefits, including:

1. No external dependencies on web servers and other 3rd party products.
2. A mature framework with support for different message and transport types, built in security and advanced features such as [MTOM encoding](#) for large attachments.
3. And most importantly, all functionality is exposed via standard HTTP based Web Service requests.

This last point about requests being HTTP based is very important as it allows the Conversion Service to be scaled across multiple servers using standard hardware or software based load balancers, including the free NLBS that ships with Windows. By utilising a load balanced environment you can achieve linear scalability and automatic failover.



*Example based on SharePoint Front ends, but Java and .NET deployments work the same.*

For details about tuning the various options of the Muhimbi Conversion Service, and installation in general, see the *Resources* section at the end of this appendix.

## Single Server Farm

The most basic configuration possible is to install everything on a single server. Just follow Chapter 2 in this Administration Guide (including all links to Appendices) and you are ready to go. There is nothing else to configure and, if needed, the Web Service can be accessed on the following URL:

`http://localhost:41734/Muhimbi.DocumentConverter.WebService/`

## Small farms with a single Conversion Service

For slightly larger deployments of 2 or more servers, but with a single conversion server, deployment is very simple as well. Let's take the following example:

- Server 1: A new or existing Application Server that will run the Muhimbi Conversion Service.
- Server 2: A server that will run either a SharePoint WFE or a custom solution.

In this particular case it is a matter of installing the Conversion Service on Server 1 as per the instructions in Chapter 2 of this Administration Guide. No further changes are required, but remember that the web service URL is now:

`http://Server1:41734/Muhimbi.DocumentConverter.WebService/`

Naturally you will need to replace 'Server1' with the server's host name.

In case of a SharePoint deployment you need to deploy the WSP file to Server 2 and enter the Web Service URL in the PDF Converter's SharePoint Central Administration screen. If you are not running SharePoint, but your own Java / .NET / etc solution then you will need to pass the above mentioned web service URL using whatever syntax is common for your platform.

### Large farms with multiple conversion servers

More complex environments that require a high level of scalability and the ability to fail over servers usually utilise multiple front end servers and multiple conversion servers. For example:

- Server 1: A new or existing Application Server that will run the Muhimbi Conversion Service.
- Server 2: A new or existing Application Server that will run the Muhimbi Conversion Service.
- Server 3: A server that will run either a SharePoint WFE or a custom solution.
- Server 4: A server that will run either a SharePoint WFE or a custom solution.
- Load Balancer

In this scenario the Conversion Service will need to be installed on Server 1 and Server 2 as per the instructions in Chapter 2 of this Administration Guide. Once installation is complete the web service can be reached on the following 2 URLs:

```
http://Server1:41734/Muhimbi.DocumentConverter.WebService/  
http://Server2:41734/Muhimbi.DocumentConverter.WebService/
```

Although in theory you could build functionality into your software to alternate requests between these two URLs, it is much easier and more robust to use an off-the-shelf HTTP load balancer or Windows NLBS. How this works in detail differs per load balancer, but it usually involves creating a *virtual host* and configure this virtual hosts to send requests to *Server1* and *Server2*. In this example we assume that this virtual host is named *LoadBalancer*, resulting in the following Web Service URL:

```
http://LoadBalancer:41734/Muhimbi.DocumentConverter.WebService/
```

In case of a SharePoint deployment you need to deploy the WSP file to Server 3 or 4, it doesn't matter which one, the installer will automatically deploy it to all WFEs.

The *LoadBalancer* URL needs to be added to the PDF Converter's SharePoint Central Administration page, or your application's configuration file. When a request is made to the load balanced URL the load balancer will automatically detect which of the servers are available and only send requests to servers that are up and running. If multiple servers are up and running it will distribute requests between all servers.

If you have multiple SharePoint Web Front End servers it may be tempting to install a copy of the Conversion Service on each WFE. Although this is a supported scenario we recommend deploying the Conversion Service on separate Application Servers to make sure that all your WFE's resources are available for running SharePoint, which can be quite resource hungry by itself.

### Licensing

Please make sure you are licensed for the correct number of servers. Muhimbi's software is licensed on a 'per server' basis. This means that you need a license for every server that runs our software including all Web Front End servers in your SharePoint farm and all servers running the conversion service.

For more details see this [Knowledge Base Article](#).

### Resources

For more details about deployment and configuration see the following resources:

- [Web Services related topics in the Knowledge base](#).
- [Deployment related topics in the Knowledge base](#).

## Appendix - Creating Custom Converters

The *Muhimbi PDF Conversion Service* allows custom converters to be added with relative ease. This is useful for converting file types specific to your organisation or for file types that have not (yet) been implemented by Muhimbi in the main product.

The following example describes how to replace the existing MS-Word converter, which relies on MS-Word being present, with a simple third party converter that works well enough for simple documents. Some programming knowledge is required.

**Update:** *It is also possible to use 3<sup>rd</sup> party executables as custom converters without any programming, see Error! Reference source not found..*

The steps are as follows:

1. Create a new Visual Studio project, select *Class Library (C#, .net 3.0)* as the template and give the project an appropriate name, e.g. *CustomConverters*.
2. Add references to the following DLLs. They are located in the folder the Muhimbi Document Conversion Service has been installed in, usually *c:\Program Files\Muhimbi....*
  - Muhimbi.dll
  - Muhimbi.DocumentConverter.WebService.dll
  - Muhimbi.DocumentConverter.WebService.Data.dll
  - System.Runtime.Serialization (Add reference from the .NET tab)
3. Change your project's Assembly name and default namespace to something sensible, e.g. *Muhimbi.DocumentConverter.WebService.CustomConverters*. This can be done by *right-clicking* on your project and selecting *Properties*. The *Application* tab allows these settings to be changed.
4. Delete the automatically generated *class1.cs* file and add a new class named *WordConverter.cs*. **Make sure the class definition is *public*.**
5. Inherit *Muhimbi.DocumentConverter.WebService.AbstractDocumentConverter* in the *WordConverter* class and implement the members (*right-click* on the base class name and select *Implement Abstract Class*).
6. Add the following 2 constructors and make sure they call the base constructors.

```
public WordConverter() : base()
{

}

public WordConverter(Stream sourceFile, OpenOptions openOptions,
                    ConversionSettings conversionSettings)
    : base(sourceFile, openOptions, conversionSettings)
{

}
```



- Next up, we need to implement the *RunDiagnostics* method. This method is normally used to carry out an internal end-to-end conversion to verify that the converter and all prerequisites have been installed correctly. In this test we simply return a new *DiagnosticResultItem* with the *Valid* property set to *true*.

```
public override DiagnosticResultItem RunDiagnostics()
{
    DiagnosticResultItem dri = new DiagnosticResultItem();
    dri.Valid = true;
    return dri;
}
```

- If we need to look further than just the file extension to determine the file type then we can optionally override the *CanConvert* method and look inside the stream (available in the *\_sourceFile* member variable). This is not necessary for this sample converter, but an example is provided below.

```
public override bool CanConvert(string[] fileExtensions)
{
    // ** Do we know anything about this extension
    if (base.CanConvert(fileExtensions) == false)
        return false;

    // ** Investigate in more detail
    ...implement your own...
}
```

- The next and final method to implement is the actual *Convert* method, which is where all the magic happens. As it is not feasible to develop an MS-Word to PDF converter from scratch, the sensible approach to take is to use a third party library such as [SyncFusion DocIO](#) or [Aspose.Words](#) (download the archive that contains just the DLLs). In this example we are going to use Aspose's library for processing MS-Word files. It is not perfect, but for some documents such as forms and simple text documents it works very well.

Copy *Aspose.Words.dll* into the project directory and add a reference to it. Copy the following code into the *WordConverter* class.

```
public override Stream Convert()
{
    try
    {
        // ** Validate as certain options are not supported by this converter
        if (_openOptions.AllowMacros != MacroSecurityOption.None)
            Logger.Warn("Macros are not supported by this converter.");

        // ** Set the licences for Aspose.Words.
        Aspose.Words.License wordLicence = new Aspose.Words.License();
    }
}
```



```
//wordLicence.SetLicence("Enter your license in here.");

Document asposeDocument = new Document(_sourceFile, null,
                                         LoadFormat.Auto, _openOptions.Password);

// ** Do we need to refresh the fields etc?
if (_openOptions.RefreshContent == true)
    asposeDocument.Range.UpdateFields();

// ** Convert the Document to PDF and save it as a memory stream.
if (_conversionSettings.Format == OutputFormat.PDF)
{
    MemoryStream convertedStream = new MemoryStream();
    PdfOptions options = new PdfOptions();
    // ** Specify the PDF Profile
    if (_conversionSettings.PDFProfile == PDFProfile.PDF_1_5)
        options.Compliance = PdfCompliance.Pdf15;
    else
        options.Compliance = PdfCompliance.PdfA1b;

    // ** How to deal with bookmarks
    if (_conversionSettings.GenerateBookmarks ==
        BookmarkGenerationOptionAutomatic)
        options.HeadingsOutlineLevels = 9;
    else if (_conversionSettings.GenerateBookmarks ==
        BookmarkGenerationOptionCustom)
        options.BookmarksOutlineLevel = 9;

    // ** Correct the start and end pages if needed
    int startPage = _conversionSettings.StartPage != 0 ?
        _conversionSettings.StartPage - 1 : 0;
    int pageCount = asposeDocument.PageCount - startPage;
    if (_conversionSettings.EndPage != 0)
        pageCount = Math.Min(_conversionSettings.EndPage - startPage,
                             pageCount);

    // ** Carry out the actual conversion
    asposeDocument.SaveToPdf(startPage, pageCount, convertedStream,
                             options);
    return convertedStream;
}
else
{
    throw new NotSupportedException("Outputformat '" +
                                    _conversionSettings.Format +
                                    "' not supported by this Converter.");
}
}
catch (UnsupportedFileFormatException ex)
{
    throw new WebServiceInternalException(
        WebServiceExceptionType.FileFormatNotSupported, ex.Message);
}
catch (Exception ex)
{
    string message = "An error occurred while converting a file";
    if (_openOptions != null && _openOptions.OriginalFileName != null)
        message += " - " + _openOptions.OriginalFileName;
    Logger.Error(message, ex);
    throw new WebServiceInternalException(
        WebServiceExceptionType.InternalError, message);
}
}
```

10. Compile the project and copy the output DLL as well as *Aspose.Words.dll* to the directory that holds the *Muhimbi Document Conversion Service*.

11. Edit the service's config file and make the following changes:

- If the file extensions for the new converter are currently handled by a different converter then remove these extensions from the existing converter.
- Add the definition for the new converter to the config file as per the following example. For details see 2.4.6 *Adding custom converters / changing file extensions*.

```
<add key="CustomWordConverter"
      description="Custom MS-Word Converter"
      fidelity="Full"
      supportedExtensions="doc,docx"
      type="Muhimbi.DocumentConverter.WebService.CustomConverters.WordConverter,
           Muhimbi.DocumentConverter.WebService.CustomConverters,
           Version=1.0.0.0, Culture=neutral, PublicKeyToken=null" />
```

12. Restart the service to activate the changes.

```
Net stop "Muhimbi Document Converter Service"
Net start "Muhimbi Document Converter Service"
```

13. Finally test if everything is working correctly, either from:

- **SharePoint:** Open *Central Administration / Application Management / Muhimbi Document Converter Settings* (In SharePoint 2010 this screen is located in *Central Admin / General Application Settings / Muhimbi Document Converter Settings*), verify that the new converter is added to the list, check the tick box and click *Validate Settings*. If everything is working correctly then don't forget to save the changes using the *OK* button.
- **Winforms Diagnostics Tool:** Launch the *Diagnostics Tool* from the Windows Start Menu, navigate to the *WS Diagnose* Tab and click the *Request Diagnostics* button. Verify the new converter is listed and *Valid = True*.

Any errors are logged to the Windows Application Event Log.

Congratulations, you have created your first custom converter. Source code for *WordConverter.cs* and the latest version of this tutorial can be found on the [Muhimbi website](#).

### Exception handling

Although you can let exceptions bubble up, we recommend catching any exceptions, inspecting the root cause of the problem and then throwing a specific *WebServiceInternalException* using one of the following exception types.

```
public enum WebServiceExceptionType
{
    /// <summary>
    /// Unknown error
    /// </summary>
    Unknown,
    /// <summary>
    /// File format not supported
    /// </summary>
    FileFormatNotSupported,
    /// <summary>
    /// File corrupt
    /// </summary>
    CorruptDocument,
    /// <summary>
    /// An error occurred while opening the file
    /// </summary>
    ErrorOpeningFile,
    /// <summary>
    /// Conversion process timeout
    /// </summary>
    ConversionTimeOut,
    /// <summary>
    /// Application hang. Can happen when document is password protected
    /// </summary>
    ConverterNotResponding,
    /// <summary>
    /// The underlying converter has not been installed or not correctly installed.
    /// </summary>
    ConverterNotInstalled,
    /// <summary>
    /// Internal Validation (Should only happen during development)
    /// </summary>
    InternalError,
    /// <summary>
    /// The specified output format is not supported (e.g. XPS output for HTML
    /// conversion)</summary>
    OutputFormatNotSupported,
    /// <summary>
    /// Configuration file is invalid, e.g. no steps defined in a multi step converter
    /// </summary>
    ConfigurationError,
    /// <summary>
    /// The trial has expired (e.g. when processing non-PDF files)
    /// </summary>
    TrialExpired
}
```

## Appendix - Invoke 3<sup>rd</sup> party Converters

The PDF Converter has had the ability to add custom converters for a while (See *Appendix - Creating Custom Converters*). However, although these plug-ins work very well, if you are not a developer, or you are not familiar with .net based development, then implementing a custom converter may be less than trivial.

As of version 6.1 it is possible to use existing command line based 3<sup>rd</sup> party conversion engines using the new *Command Line Converter*.

*The latest version, and further details, of this topic can be found [in this post on our Blog](#).*

The Command Line Converter is very simple to setup as you can see in the following config file fragment for Siemens Teamcenter:

```
<add key="CommandLineConverter"
description="Converts using 3rd party executables"
fidelity="Full"
supportedExtensions="hp,hpg,hpgl,hpgl2,plt,cgm,m1r"
supportedOutputFormats="pdf"
type="Muhimbi.DocumentConverter.WebService.CommandLineConverter,
Muhimbi.DocumentConverter.WebService, Version=1.0.1.1, Culture=neutral,
PublicKeyToken=c9db4759c9eaad12"
parameter="c:\splm\vis20072\VVCP\prepare.exe | -PDF {0} -combine -page all
-out {1} -overwrite -size {Parameter1}"/>
```

The parameters are as follows:

1. **key:** Give the converter a unique name in case multiple Command Line Converters are in use. Otherwise accept the default value.
2. **description:** The description of the converter exposed via the web services API. As, for example, displayed in our SharePoint Central Administration screen.
3. **supportedExtensions:** A list of input file formats supported by this converter.
4. **supportedOutputFormats:** The file formats the converter can generate. This is not limited to just PDF as our software fully supports cross-conversion.
5. **type:** Do not touch.
6. **parameter:** The location of the 3rd party converter and the arguments to send to it. Note that these 2 values are separated by a vertical pipe character '|'. The arguments section supports the following parameters:
  - **{0}:** This will automatically be replaced with the full path to the input file.
  - **{1}:** This will automatically be replaced with the full path and filename where the output file should be generated.

- **{Parameter1}, {Parameter2}....{Parameter10}**: Optional parameters that can be passed in via the web services interface using *ConverterSpecificSettings\_CommandLineConverter*. Use this to pass proprietary information to the third party converter such as page size or special processing instructions.

Once everything has been configured, all mapped file formats will be picked up automatically and treated exactly the same as all other file formats supported by the Muhimbi PDF Converter.

## Appendix - Licensing

All Muhimbi products are licensed in a way that allows maximum flexibility. Please familiarise yourself with the licensing agreement, particularly section 3 – *Grant of License*, before purchasing our software.

For details see:

1. How we license:  
<http://blog.muhimbi.com/2010/01/how-we-license-our-products-make-sure.html>
2. The License Agreement:  
<http://www.muhimbi.com/Software-License-Agreement.aspx>
3. Licensing FAQ:  
<http://www.muhimbi.com/support/faqs.aspx#Licensing>

In summary we support the following license types. Please refer to the resources above for exact details:

	Number of servers	Unlimited Web Applications	Unlimited Web Farms	Unlimited Locations	Redis-tributable	Includes Source code
Server License	1	<input checked="" type="checkbox"/>				
Enterprise License	Unlimited	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
OEM License	Unlimited	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Source Code License	Unlimited	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Please note that some older SharePoint specific license types have been discontinued. However, these are still valid for those customers that have purchased them in the past. Please see the License Agreement for details about these old style licenses.

1. **Free evaluation version:** If you install the software without a license you are using the evaluation version. The software is fully functional without any time limits, but an evaluation message will be displayed on most screens, in the workflow history and in any generated document. Please do not use any evaluation software in your production environment. You can get support using any of the means in the Support area on our site.
2. **Server License:** The easiest way to license our software is to buy a Server License for each of your servers, virtual or physical, that runs our software. This could be all SharePoint servers in your farm or non SharePoint based servers that run our software.

3. **Enterprise License:** If you are running our software on more than a handful of servers then it may be more economical to purchase an Enterprise License. This allows the software to be installed on an unlimited number of servers in the organisation.
4. **OEM License:** If you wish to bundle our software with your own solution and redistribute it to 3<sup>rd</sup> parties then you require an OEM License. Please read the details in the Software License Agreement for more information.
5. **OEM License + Source Code:** If you need all the benefits of the OEM License and / or you need access to the source code to make modifications specific to your organisation, then this license type is the best option. Note that we do not provide support for our software once changes have been made to the source code. Please read the details in the Software License Agreement if you want to bundle our software with your own solution.

*The PDF Converter for SharePoint license is limited to use from SharePoint environments only. If you wish to invoke the PDF Conversion Service from a non-SharePoint based environment, e.g. Java, .NET or any other Web Services capable system then you will need to purchase a license for the PDF Converter Services.*

*The PDF Converter Professional license is an add-on that adds additional functionality to either the PDF Converter for SharePoint or the PDF Converter Services. This functionality, e.g. PDF/A post-processing, is usually associated with more complex environments and has therefore been separated from the main product. Please note that the PDF Converter Professional is a license that must be applied alongside a valid license of the PDF Converter for SharePoint or PDF Converter services. A separate download of the Professional version of the software is not needed, the license unlocks all functionality.*