



## **Print2PDF Version 9**

**Installation of the Web Service and Web  
Application**

# Contents

<b>1.</b>	<b>Introduction .....</b>	<b>4</b>
<b>2.</b>	<b>Description and Terminology .....</b>	<b>4</b>
<b>2.1.</b>	<b>Server Installation Components .....</b>	<b>4</b>
2.1.1.	Print2PDF Web Service .....	4
2.1.2.	Print2PDF Web Application .....	4
2.1.3.	Print2PDF Service .....	5
2.1.4.	Print2PDF Application .....	5
<b>2.2.</b>	<b>Features Available in Different Versions.....</b>	<b>6</b>
<b>3.</b>	<b>Web Service Integration.....</b>	<b>7</b>
<b>3.1.</b>	<b>Basic Operations of the Web Service .....</b>	<b>7</b>
3.1.1.	Conversion into the PDF Format.....	7
3.1.2.	Time Stamp .....	9
3.1.3.	Test Operations .....	10
<b>4.</b>	<b>Web Service and Web Application Configuration .....</b>	<b>12</b>
<b>4.1.</b>	<b>Web Service Configuration .....</b>	<b>12</b>
4.1.1.	TSA Configuration .....	12
4.1.2.	Conversion Configuration .....	13
4.1.3.	Thread Number .....	13
4.1.4.	Web Application.....	13
<b>5.</b>	<b>Installation.....</b>	<b>14</b>
<b>5.1.</b>	<b>Selecting the Right Version of Service and Application .....</b>	<b>14</b>
<b>5.2.</b>	<b>Installation Instructions .....</b>	<b>14</b>
5.2.1.	Upgrading an Existing Installation .....	14
5.2.2.	New Installation.....	14
<b>5.3.</b>	<b>Basic IIS Configuration .....</b>	<b>15</b>
5.3.1.	Preparation of the Application Pool .....	15
5.3.2.	BInstallation Process .....	17
5.3.3.	ASP.NET 1.1 Version .....	17
5.3.4.	ASP.NET 2.0 and Higher Version .....	20
<b>5.4.</b>	<b>Installation of the Windows Service.....</b>	<b>21</b>
<b>6.</b>	<b>Use of the Service.....</b>	<b>23</b>
<b>6.1.</b>	<b>Configuration of the Service for TSA .....</b>	<b>23</b>
<b>6.2.</b>	<b>Converting Microsoft Office and Open Office Documents Using the Web Service.....</b>	<b>26</b>
6.2.1.	Example of a DCOM Configuration.....	26



- 6.2.2. Converting OpenOffice Files Using the Web Service ..... 29
- 7. Appendices ..... 31**
- 7.1. Appendix 1: WSDL Service Definition File.....31**
- 7.2. Appendix 2: Error Codes for the ConvertFile Function .....40**

# 1. Introduction

This manual describes the different installation options for the web service and web application distributions of Print2PDF 9.0 on different (MS Windows based) operating systems, the different installation configuration options and examples of their use.

This manual assumes that its user has administrator-level knowledge of the installation and use of IIS and ASP.NET, DCOM configurations and the creation and administration of user accounts.

## 2. Description and Terminology

The latest Print2PDF distribution includes executable files for the related web service and web application in versions for ASP.NET 1.1 (32-bit version only) and for ASP.NET 2.0 and higher (32- and 64-bit versions).

The files can be found in the following directories:

...Software602\Print2PDF\WebApp and ...Software602\Print2PDF\WebService for the ASP.NET 1.1 version;

..\WebApp\_20 and ...\WebService\_20 for the ASP.NET 2.0 and higher versions

or in the \WebApp\_x64 and WebService\_x64 directories for the 64-bit version.

### 2.1. Server Installation Components

This manual uses the following names for the different components of Print2PDF:

- Print2PDF Web Service
- Print2PDF Web Application
- Print2PDF Service
- Print2PDF Application


#### 2.1.1. Print2PDF Web Service

The Print2PDF Web Service is a software agent designed to perform the conversion of files into a PDF format using the SOAP protocol and to issue a provider timestamp. The contents of the source file being converted to the PDF format are converted to the Base64 encoding scheme. Other input data for the conversion are the filename (which determines the conversion method to be used) and the conversion profile (configuration of conversion parameters). The results of the conversion are a PDF file (once again, encoded using Base64) with a name for the converted file (\*.pdf), a timestamp and, occasionally, an error message.

The web service distribution consists of the following files, which are part of the distribution directory: the Bin folder contains the actual Print2PDF\_service.dll file and the directory also contains the default.asmx file, the web.config configuration file and the Global.asax file. The PipeSrv.exe executable file is also part of the distribution package.

#### 2.1.2. Print2PDF Web Application

The Print2PDF Web Application is a demo application, which shows how the Print2PDF Web Service is used, and it allows the user to test the web service after it has been installed.



The distribution package once again consists of the actual Print2PDF Application\_WebApp.dll file containing the application (it can be found in the Bin folder), the web.config configuration file, other items (icons, logo) and multiple language versions of the source documentation. Besides the basic web form (Default.aspx) in English, the distribution also includes other forms. Unlike in the Print2PDF Application (desktop version), the installation does not use the operating system running on the given device to determine which version of the documentation will be used; but, instead, it makes the determination based on a specific document, which is entered during the IIS configuration.

### 2.1.3. Print2PDF Service

The Print2PDF Service is a Windows service that performs a conversion on a server. It is also used to handle other automated conversion tasks such as folder tracking (see the Print2PDF manual).

### 2.1.4. Print2PDF Application

The Print2PDF Application is a basic program that allows the user to convert documents into the PDF format. The application is actually a virtual printer driver, which converts files by 'printing them' in any application capable of opening the given file.

**Note:** *This means that there are certain limitations on the use of the Print2PDF server installation. In order to perform the conversion, the user must have access to a suitable application, which can be used in an automated regime. This is a requirement that not all applications meet (e.g. some versions of OpenOffice), or it is not recommended for various reasons (e.g. Microsoft Office).*

The files that are part of the Print2PDF Application package are installed in the appropriate program directory (program files) and also in the Spool/ Drivers/ directory (this can vary depending on the OS being used).

## 2.2. Features Available in Different Versions

Feature	ASP.NET 1.1	ASP.NET 2.0 and higher	
		32-bit version	64-bit version
BeginAppendSession	-	+	+
ConvertFile	+	+	+
ConvertFileAppend	-	+	+
EndAppendSession	-	+	+
GetProfileList	+	+	+
GetTimeStamp	+	+	+
TestTimeStampService	+	+	+
Test_ConvertFile_DOC	-	+	+
Test_ConvertFile_TXT	+	+	+
Test_ProfileList	+	+	+
Service IP Address Change (default = localhost)	-	+	+

## 3. Web Service Integration

To integrate the Print2PDF\_service web service into your application, use the web service interface definition ([Appendix 1](#)).

### 3.1. Basic Operations of the Web Service

#### 3.1.1. Conversion into the PDF Format

Basic functions: **GetProfileList** and **ConvertFile**.

GetProfileList obtains a list of available profiles (identification and name) from the server and you can then choose the profile that is to be used for the ConvertFile operation.

##### 3.1.1.1. Example of GetProfileList Call and Response

**The GetProfileList operation is initiated without parameters:**

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:prin="http://software602.com/print2pdf/">
  <soapenv:Header/>
  <soapenv:Body>
    <prin:GetProfileList/>
  </soapenv:Body>
</soapenv:Envelope>
```

**Server response:**

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetProfileListResponse xmlns="http://software602.com/print2pdf/">
      <GetProfileListResult>0</GetProfileListResult>
      <Profiles>
        Profile.ini
        Standard
        Profile001.ini
        Confidential
        Profile002.ini
        Minimum File Size
        Profile003.ini
        High Resolution Print
        Profile004.ini
        Archive (PDF/A-1a)
      </Profiles>
      <Report/>
    </GetProfileListResponse>
  </soap:Body>
</soap:Envelope>
```

The above example shows the syntax used in the server response: the operation returns a list of profiles in two strings – the first one contains the identification and the second one contains the name of the profile. The identification profile is used as an input for the web service.

### 3.1.1.2. Example of ConvertFile Call and Response

The ConvertFile operation includes mandatory parameters:

- **Input:** The source file being converted (base64).
- **Profile:** Profile ID (string) – if not specified or not available on the server, the default profile (from the configuration) will be used for the conversion.
- **FileName:** The name of the file being converted – determines the use of a conversion script.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://software602.com/print2pdf/">
  <soap:Body>
    <tns:ConvertFile>
      <tns:Input>Tm93IGlzIHRoZSB0aW11IGZ
      ....
      vciBhbGwgZ29vZCBtZW4uLi4uLg==</tns:Input>
      <tns:FileName>c:\test.txt</tns:FileName>
      <tns:Profile>Profile001.ini</tns:Profile>
    </tns:ConvertFile>
  </soap:Body>
</soap:Envelope>
```

#### Server response:

The response consists of a PDF file (sent as a data stream) or an error message.

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <ConvertFileResponse xmlns="http://software602.com/print2pdf/">
      <ConvertFileResult>0</ConvertFileResult>
      <Output>JVBERi0xLjMNCiX/////DQoxIDAgb2JqDTw8DS9GaWVsZHMgW10NL0.....
      NTY0Njg0QTQ20TAzREYyRTFCMQQ+PEM5RkJDNkFDQjBBRjU2NDY4NEE0NjkwM0RGMkUxQjNEPl0N
      Pj4Nc3RhcncR4cmVmDTM5MjQ3DSU1RU9GDQ==</Output>
      <Report>Hotovo</Report>
    </ConvertFileResponse>
  </soap:Body>
</soap:Envelope>
```

ConvertFileResult – the result of the conversion (see [Appendix 2](#)).

### 3.1.1.3. Other Conversion Operations

The following are other operations that are part of the conversion process:

- **BeginAppendSession,**
- **ConvertFileAppend,**
- **EndAppendSession.**

These operations are used to combine multiple files into a single file in the PDF format.

These operations must always be run in the following sequence: BeginAppendSession, ConvertFileAppend ... (for every file being converted) and (**as the last operation**) EndAppendSession.

**BeginAppendSession** is a parameter-free operation, which is used to indicate that the following files are to be combined during the conversion.

```
<?xml version="1.0" encoding="utf-8"?> <soap:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body>
<BeginAppendSession xmlns="http://software602.com/print2pdf/" />
</soap:Body> </soap:Envelope>
```

**ConvertFileAppend** is an operation identical to ConvertFile. The only difference is that this operation includes the ID parameter, which identifies the group of files that are to be combined (all of them having the same ID).

```
<?xml version="1.0" encoding="utf-8"?> <soap:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body>
<ConvertFileAppend xmlns="http://software602.com/print2pdf/">
<Input>base64Binary</Input> <FileName>string</FileName>
<Profile>string</Profile> <AppendSessionID>int</AppendSessionID>
</ConvertFileAppend> </soap:Body> </soap:Envelope>
```

**EndAppendSession** is an operation, which concludes the conversion of the files that are being combined. The operation has just a single parameter – an ID.

```
<?xml version="1.0" encoding="utf-8"?> <soap:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body>
<EndAppendSession xmlns="http://software602.com/print2pdf/">
<SessionID>int</SessionID> </EndAppendSession> </soap:Body> </soap:Envelope>
```

### 3.1.2. Time Stamp

The **GetTimeStamp** operation is used to obtain a time stamp.

```
POST /WebService_x64/Default.aspx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://software602.com/print2pdf/GetTimeStamp"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetTimeStamp xmlns="http://software602.com/print2pdf/">
      <Hash>string</Hash>
```

```

    <Flags>long</Flags>
  </GetTimeStamp>
</soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetTimeStampResponse xmlns="http://software602.com/print2pdf/">
      <GetTimeStampResult>string</GetTimeStampResult>
      <Error>string</Error>
    </GetTimeStampResponse>
  </soap:Body>
</soap:Envelope>

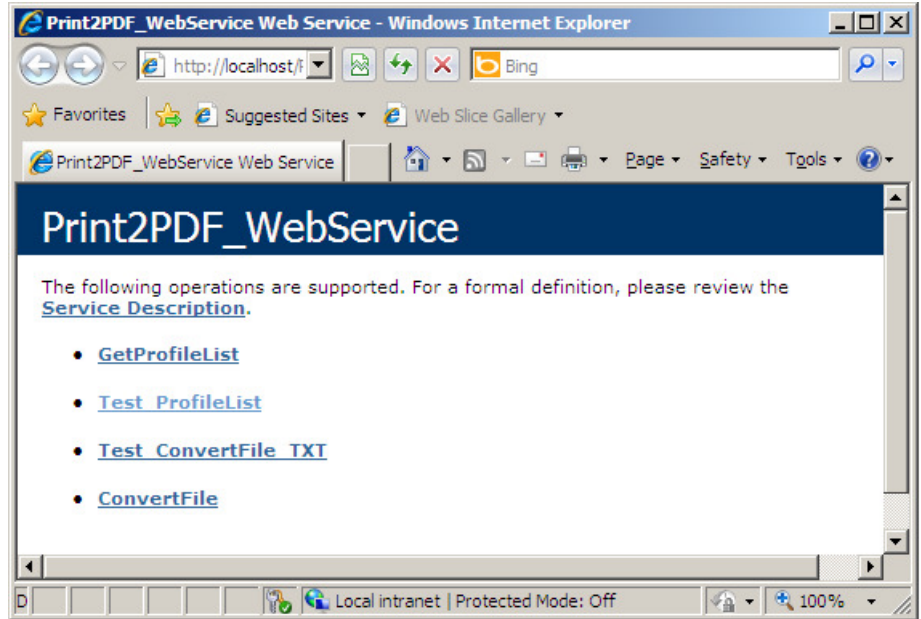
```

The source used for this operation is the hash of the document for which the time stamp is being obtained and the flags attribute (this is reserved for future use, the current default = 0).

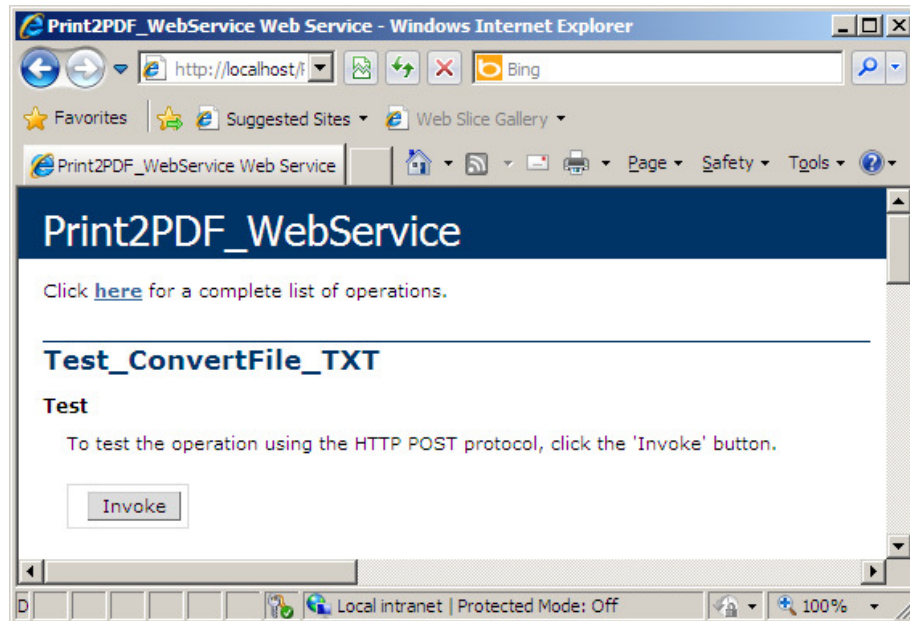
The output is an encoded time stamp (GetTimeStampResult - BASE-64 in the TimeStampResp encoded structure, see RFC 3161) and a return value with a description of an error (if applicable).

### 3.1.3. Test Operations

These operations are only used to test the functionality of the installed web service. The operations can only be run on the computer on which the service has been installed and they can be accessed through a link to the given operation located on the home page of the web service.



Clicking on the link for the desired operation will bring up a page with the **Invoke** button.



Clicking on this button will submit a request for the execution of the given test operation. The result of this process is a message, which will appear as a new browser window (tab) saying either that the operation has been successful or describing the error that occurred during the operation.

- **GetProfileList** – This function will return a list of current profiles.
- **TestTimeStampService** – Connects to the specified TSA server and returns a time stamp.
- **Test\_ConvertFile\_DOC** – Converts a predefined document in the MS Word format (this feature may require a change of access rights to the MS Word document DCOM object).
- **Test\_ConvertFile\_TXT** – Converts a predefined text document.

## 4. Web Service and Web Application Configuration

The configuration is performed using the web.config files located in the directories containing the files for the web service and the web application.

There is also a "web.config.sample" file that contains a sample (demo) configuration and the sample settings for certain elements. This file can be used as a template for editing. After the desired parameter changes have been made the file can simply be saved as "web.config".

### 4.1. Web Service Configuration

Besides the standard parameters (described in ASP.NET documentation – for an example, see: <http://msdn.microsoft.com/en-us/library/ff361664.aspx>), the Print2PDF\_service uses certain specific parameters. The section used for the configuration of the TSA service is a specific parameter.

#### 4.1.1. TSA Configuration

The configuration of the process of obtaining time stamps from the portal is done through the use of the AppSettings element, which contains the source address of the portal and it must also contain a specific username and password.

```
<appSettings>
  <add key="TSA_URL" value="https://tsa1-test.eidentity.cz/HttpTspServer" />
  <add key="TSA_User" value="" />
  <add key="TSA_Pass" value="" />
  <add key="TSA_CertHash" value="05 f3 3b 17"/>
</appSettings>
```

*The following keys are used to set up a proxy (if used and if the network allows it, it isn't necessary to configure access to TSA on client stations – all that is needed is access to the proxy server):*

```
<add key="Proxy_URL" value="" />
<add key="Proxy_User" value="" />
<add key="Proxy_Pass" value="" />
```

*If you want to create logs for the time stamp service, you will have to configure the access rights for the service for writing to the selected file. The TSA\_LogPath key is used to create the file.*

```
<add key="TSA_LogPath" value="C:\TSA\TSALog.log"/>
</appSettings>
```

Another option is to use a different authority for the issuance of time stamps. In order to do this, you will have to contact technical support (at: <http://support.software602.com>), where you will find out under what conditions this can be done.

Once you have met the applicable registration and activation requirements, you will be provided with a serial number, which has to be entered into the following key:

```
<add key="TSA_EnableURL" value="" />
```

You will then also have to modify the values in the TSA\_URL, TSA\_User and other keys.

The above instructions (including the one about contacting technical support) also apply to existing customers who are switching over to Version 9 from Version 8.

### 4.1.2. Conversion Configuration

Besides other settings that are part of the current configuration, it can sometimes be handy to change the maximum size limit for converted files. This can be done in the system.web section by entering the desired value for the maxRequestLength attribute.

```
<system.web>
  <httpRuntime maxRequestLength="102400" />
</system.web>
```

Insert the following line, with the desired value, into the system.web section:

```
<httpRuntime executionTimeout="360"/>
```

*Note: This can be combined with the previous settings.*

### 4.1.3. Thread Number

Add the following to the system.web section:

```
<connectionManagement>
  <add address="*" maxconnection="2">
  <add address="65.53.32.230" maxconnection="12">
</connectionManagement>
```

For more information see: <http://support.microsoft.com/kb/821268>

### 4.1.4. Web Application

**(applies to versions using ASP.NET 2.0 and higher)**

For a web application run in ASP.NET that is using a web service, you will have to specify the path to the web service in the applicationSettings section:

```
<applicationSettings>
  <Print2PDF_WebApp.Properties.Settings>
  <setting name="Print2PDF_WebApp_localhost_Print2PDF_WebService"
    serializeAs="String">
    <value>http://localhost/Print2PDF_service/Default.asmx</value>
  </setting>
</Print2PDF_WebApp.Properties.Settings>
</applicationSettings>
```

## 5. Installation

This chapter describes the methods and modifications used for the configuration files for the web service and web application based on the platform and the ASP.NET version being used.

### 5.1. Selecting the Right Version of Service and Application

The installation package contains files for the service and the application for use with ASP.NET 1.1 (32-bit version) and ASP.NET 2.0 and higher (32- and 64-bit versions).

The version for use with ASP.NET 1.1 has a limited selection of operations and the framework used in this version may experience problems when used on more recent 64-bit operating systems. The only reason this version is still being supported is to provide continuity with the previous version. It isn't guaranteed that this version will continue to be supported in future releases.

This version is recommended for use as an update for an existing installation of Version 8 of the service running on a Windows server 2003 operating system (or earlier) and IIS 6 (or earlier).

For use with an operating system running IIS 7 and 7.5, we recommend the use of the version that uses ASP.NET 2.0. In the vast majority of cases the 32-bit version will be sufficient because the Print2PDF application itself is also 32-bit software. The installation of the 64-bit version is only recommended if the target IIS server is already populated by several 64-bit application pools and the addition of another pool for a 32-bit application could cause problems. The use of the 64-bit version is also recommended if the system runs the Windows Server 2003 x64 operating system, which already contains other 64-bit web applications.

### 5.2. Installation Instructions

#### 5.2.1. Upgrading an Existing Installation

If you have already been using the Print2PDF 8 web service, you may continue to use this installation (the version for ASP.NET 1.1). It is still available with the same features, except for one important change – the TSA configuration.

A sample configuration can be found in the web.config.sample file, which can either be merged with the original file (if modified) or renamed (see above).

#### 5.2.2. New Installation

If you are doing a new installation, the following steps must be completed:

- Prepare the account under which the Windows service will run (the account must have administrative privileges and a password).
- Install and/or check the IIS settings and the ASP.NET version.
- Depending on the file formats that you plan to use in the document conversions, install the applications in which these can be opened and printed (this is normally MS Office and/or MS Visio).
- Install Print2PDF (see the user manual for instructions).
- After a system restart, prepare the directories containing the files that are part of the web service and web application. It is recommended that you select directories

containing the version that is to be installed and copy these into the folder that will be used as the source for the published IIS files. This is normally C:\inetpub\wwwroot\. After having copied the directories into this folder, you will have to check to make sure the access rights for wwwroot have been applied to the imported directories. If not, you can force the system to transfer the given privileges in the **Security / Advanced** dialog box.

- If you decide to place these files into a different location, you will have to set permissions and privileges separately for the service directory and for the application directory. The following accounts should have full access to these directories: Administrator, Internet Guest Account (IUSR), IIS Account and System. For more information see: <http://support.microsoft.com/kb/271071>.
- Install the Print2PDF Windows service using the Service Manager and launch the installed service.
- Add a new application (or virtual directory, depending on the IIS version being used) for the web service (and the web application, if desired) to the IIS manager console. For instructions, see: <http://support.microsoft.com/kb/172138>. IIS versions vary.
- You will then have to check the following settings:
  1. application name – depends on the user, but the recommended names are 'Print2PDF\_service' for the web service and 'Print2PDF' for the web application (these names are also used in the web.config file, which is part of the application).
  2. source document settings ('default.asmx' for the web service and for the web application, the document varies depending on the language version being used (e.g. for the Czech version, this is 'Default\_cs-CZ.aspx')).
  3. application pool settings – i.e. make sure that the pool is in the correct mode (32 or 64) and that it is not clashing with other applications.
  4. allow and configure ASP (for an example, see: <http://msdn.microsoft.com/en-us/library/ms178477.aspx>).
  5. check ASP.NET settings (to make sure that the right version is registered).
  6. check the IP address (depending on the IIS mode and whether the service or application is being used as the main web page or as a subdirectory).
- After the installation has been completed, it is recommended that the IIS be restarted and that the functionality of the service and application be checked (using a web browser on the computer on which the service has been installed by performing test operations). It is also recommended that the firewall settings be checked and possibly also the configuration of applications with access to the web service.
- Check the connection to the application using the web service (configuration, [link to the IP address of the service](#)).
- Perform the [test operations](#) for the service.


The system requirements and instructions for changing the IIS configuration and installing the Windows service are described below.

## 5.3. Basic IIS Configuration

### 5.3.1. Preparation of the Application Pool

Depending on the existing IIS configuration and the Print2PDF version being installed, you will have to decide whether you will need to create a new application pool or whether you can use an existing one.

A new application pool must be created if the existing pool uses a different ASP.NET version from the application being installed or if there are other factors preventing the use of an existing application pool.



Basic application pool requirements: In all cases, the Managed Pipeline Mode needs to be set to 'Classic'. To set Identity, you can either use the ApplicationPoolIdentity Network service or a specific account, depending on how the web service will be used and depending on the server settings. It is normally recommended to use ApplicationPoolIdentity or NetworkService for ASP.NET 1.1 and NetworkService for ASP.NET 2.0. A custom account must be used in certain situations such as when automatic certificate verification is required.

## 5.3.2. Installation Process

Depending on the desired configuration, install the Print2PDF service and/or application as a virtual directory – application or even as a web page.

	Print2PDF Service_service	Print2PDF Application
IIS Settings		
Directory Permissions	scripts only	
Source Document	Default.asmx	Default.aspx or other – depending on the language versions being used (e.g. Default_cs-CZ.aspx for Czech)
Authentication	allow anonymous access	
Application Pool Settings (ASP.NET 1.1 and ASP.NET 2.0 32-bit)		
Allow 32-bit Applications	Yes	
Managed Pipeline Mode	Classic	
Identity	ApplicationPoolIdentity , Network Service, custom account	
Application Pool Settings (ASP.NET 2.0 64-bit)		
Allow 32-bit Applications	No	
Managed Pipeline Mode	Classic	
Identity	ApplicationPoolIdentity, Network Service, custom account	

## 5.3.3. ASP.NET 1.1 Version

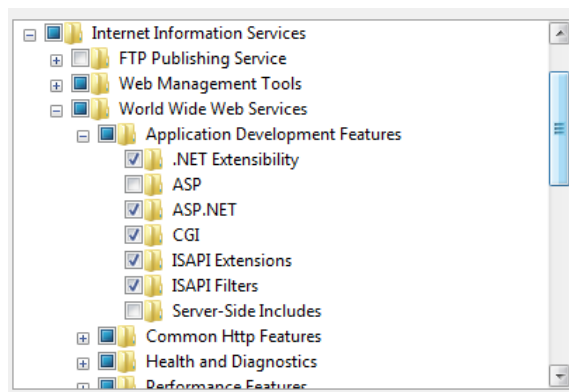
### 5.3.3.1. 32-bit Systems

#### IIS v. 5 - 6 (XP, 2003)

These systems do not require any special configuration changes and all you have to do is check that this framework has been registered for the use of IIS (aspnet\_regiis -ir command from the respective directory). ASP.NET 1.1 should either already be installed on these systems or it can be added from the original installation disk.

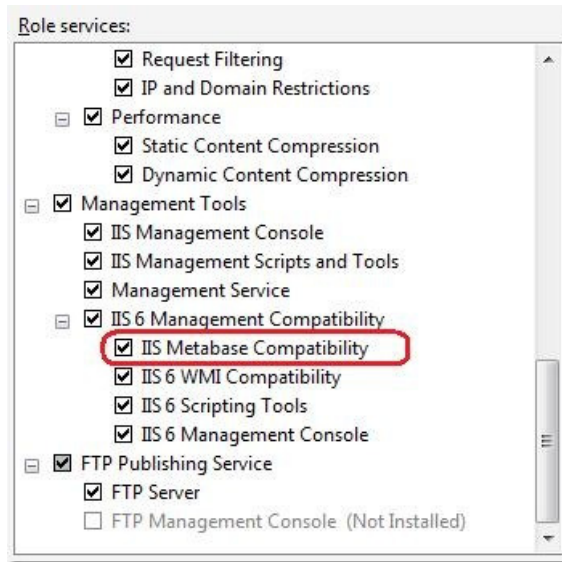
#### IIS v. 7 - 7.5 (Vista, Windows Server 2008)

The installation works, though when installing ASP.NET the system brings up an error message indicating compatibility problems. Prior to the installation, you will have to install support for ISAPI and ASP.NET.



ASP.NET 1.1 is normally not installed on these systems and you will therefore have to download it and install it.

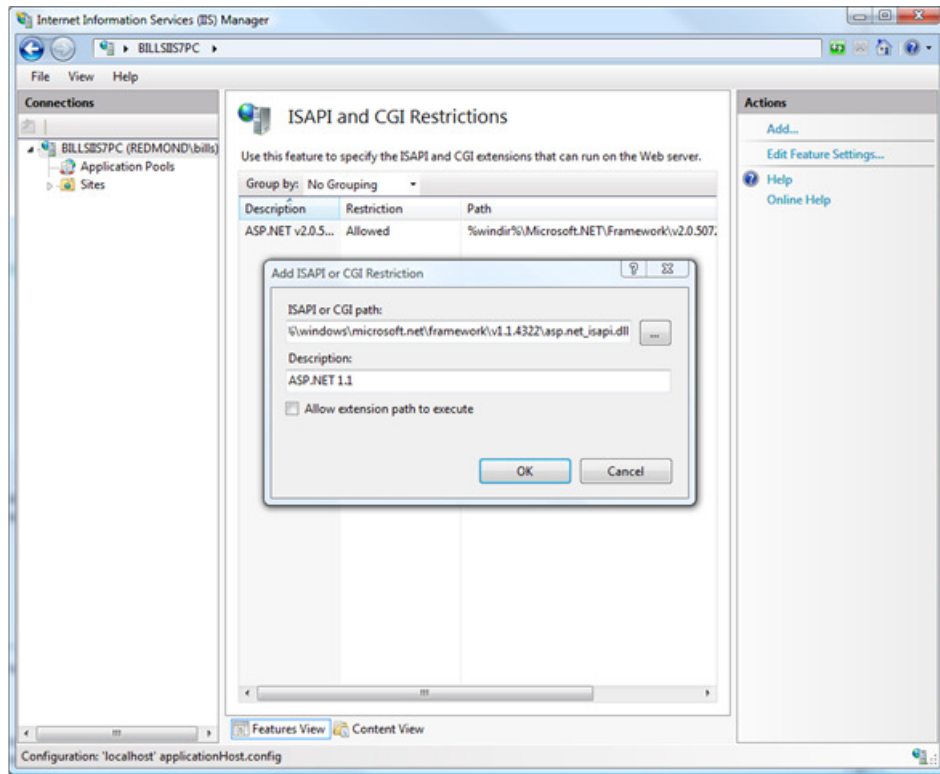
Prior to the installation, IIS7 must have the Metabase Compatibility function enabled (the check box can be found in the 'Role services' dialog box).



You will then also have to download and install Framework 1.1 and its update ([.NET Framework Version 1.1 Service Pack 1](#) and [ASP.NET Security Update for .NET Framework 1.1 SP1](#)).

Also, you will have to allow the execution of the given extension path (this is done in IIS Manager under 'ISAPI and CGI Restrictions'). If the extension is not shown on the provided list, you will have to add the following path: C:\Windows\Microsoft.NET\Framework\v1.1.4322\aspnet\_isapi.dll (or a different path based on the given installation) and allow this extension path to be executed.

Note: You can also use: <http://www.microsoft.com/web/downloads/platform.aspx>.



The following command can be used:

```
%windir%\Microsoft.NET\Framework\v1.1.4322\aspnet_regiis -enable.
```

Due to the differences between the different framework versions, you will also have to modify the machine.config configuration file (%windir%\Microsoft.NET\Framework\v1.1.4322\config\machine.config). To do this, add the following section to the <configSections> element:

```
<section name="system.webServer"
type="System.Configuration.IgnoreSectionHandler,
System, Version=1.0.5000.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" />
</configSections>
```

As the last step of preparing for the Print2PDF installation, you will have to create an application pool for ASP.NET 1.1 (e.g. by using the following command: `appcmd add apppool /name:"NewPool" managedRuntimeVersion:"v1.1"` from the following directory: %windir%\system32\inetsrv).

Recommended application pool settings:

- Managed Pipeline Mode: Classic,
- Identity: ApplicationPoolIdentity.

### 5.3.3.2. 64-bit Systems

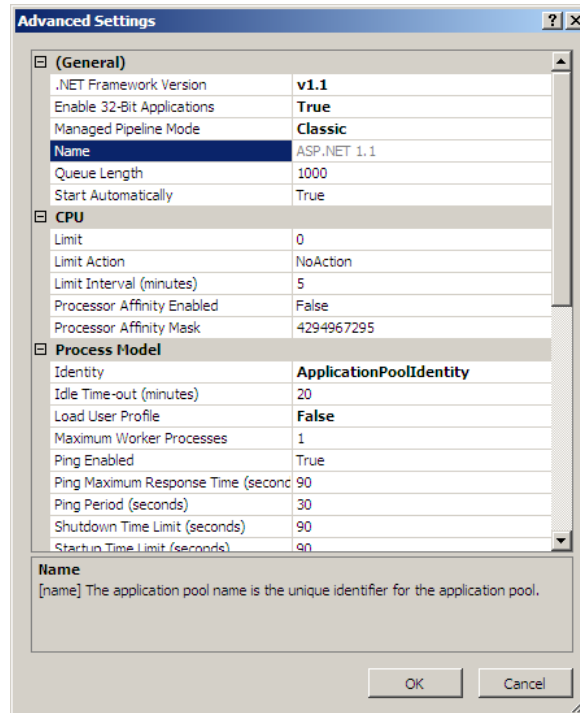
ASP.NET 1.1 only comes in a 32-bit version, which means that the 32-bit mode for IIS must be allowed.

IIS 6 (Windows Server 2003) – This system has one limitation, which is the ability to use only one mode – i.e. either 32- or 64-bit.

To enable the 32-bit mode, follow these steps:

1. from the command line, go to the following folder: %systemdrive%\Inetpub\AdminScripts

2. enter the following command: `cscript.exe adsutil.vbs set W3SVC/AppPools/Enable32BitAppOnWin64 "true"`
3. restart IIS (e.g. by typing "iisreset" in the command line).
4. IIS 7 (Windows Server 2008) – enable 32-bit applications



### 5.3.3.3. IIS Settings

This version doesn't allow a change to the IP address for the service - the local host binding remains.

### 5.3.4. ASP.NET 2.0 and Higher Version

#### 5.3.4.1. 32-bit Systems

##### IIS v. 5 - 6 (XP, 2003 )

This version is the default for these systems. Normally, all you will have to do is to check that the ASP.NET extension has been installed.

##### IIS v. 7 - 7.5 (Vista, Windows Server 2008)

On these systems, you can use both Framework 2.0, as well as versions 3.5 or 4.0 (IIS 7.5).

Recommended application pool settings:

- Managed Pipeline Mode: Classic
- Identity: Network Service (Framework 2.0), ApplicationPoolIdentity (Framework 4.0)

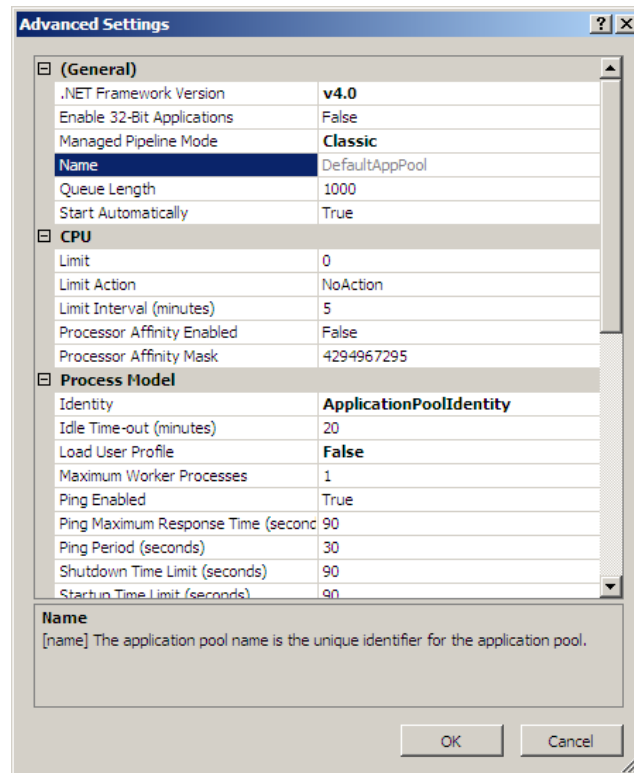
#### 5.3.4.2. 64-bit Systems

- **IIS 6** – Similar to the ASP.NET 1.1 application, you will have to enable the use of 32-bit applications for IIS.

- **IIS 7** – Similar to the 32-bit version, all you have to do is to enable the use of 32-bit applications (in Advanced Settings for the application pool, set the value for 'Enable 32-Bit Applications' to 'True'; it is also recommended that you change the launch identity to 'NetworkService'.

### 5.3.4.3. 64-bit Applications

Intended for 64-bit systems only:

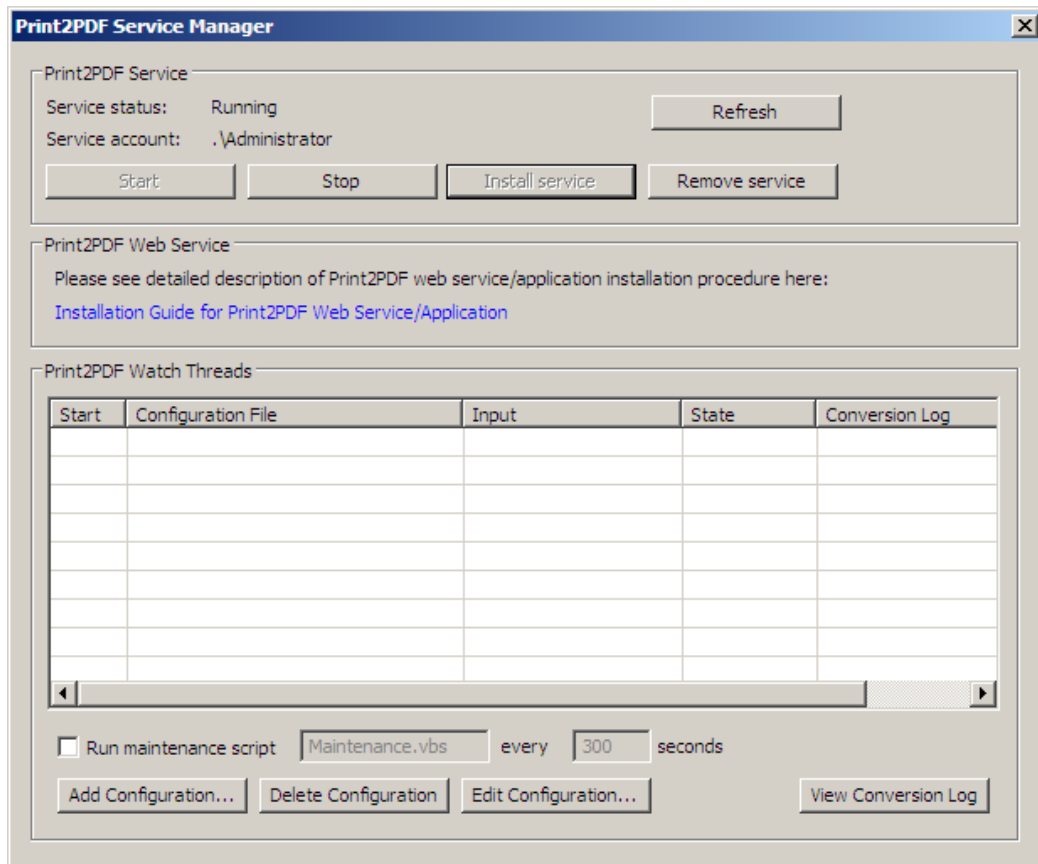


Combination of x64 service x86 application – ASP.NET Impersonation must be enabled (application settings – IIS Authentication).

## 5.4. Installation of the Windows Service

This service must be running in order to proceed with the installation of the Print2PDF web service and web application.

The service is installed from the Print2PDF Service Manager. An account with administrator privileges (password protected) is required for the installation.



Installation instructions: Clicking on **Install Service** will bring up a dialog box in which you will be asked to enter the name of an account and a password.

You can then launch the service by clicking on the **Start** button. The 'Stop' and 'Refresh' buttons are used for the administration of the installed service.

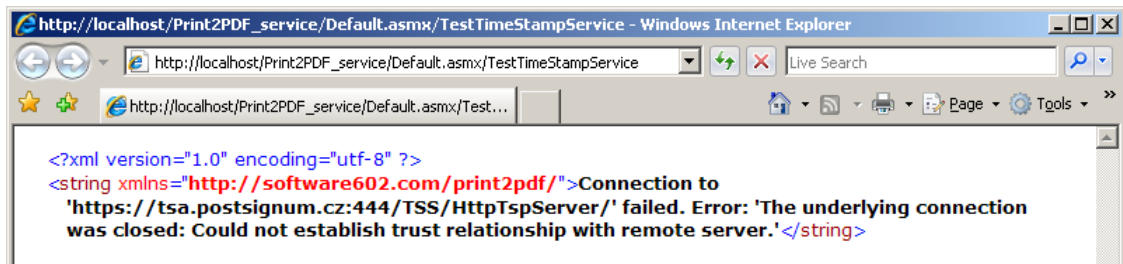
In the lower part of the dialog box, you can add configuration profiles for the tracking of multiple folders or POP3 accounts.

## 6. Use of the Service

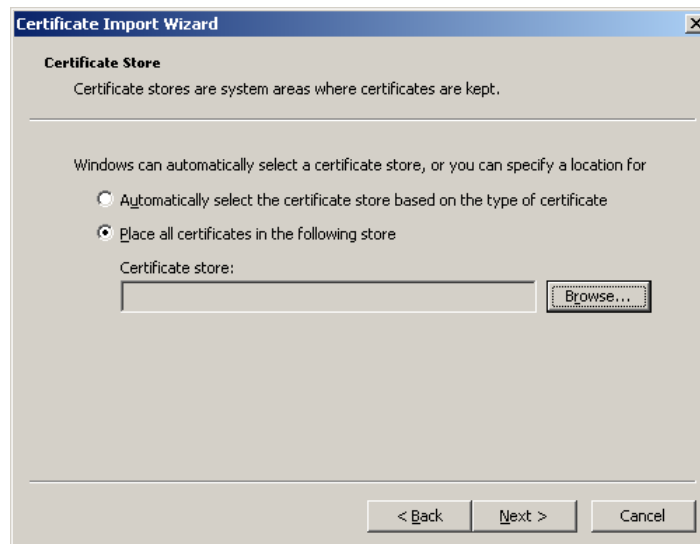
### 6.1. Configuration of the Service for TSA

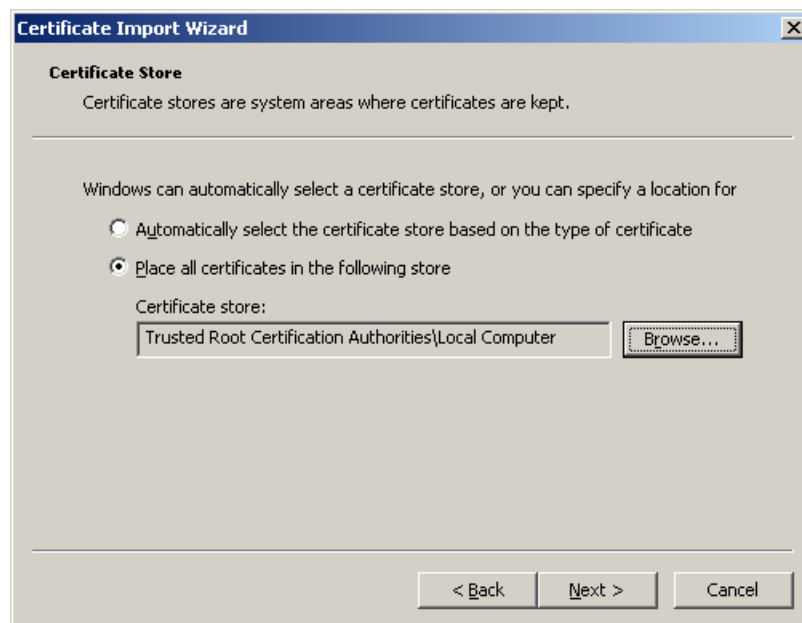
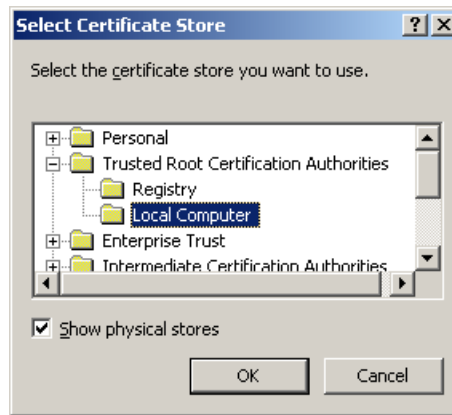
Most servers issuing time stamps are accessed through a secure connection (https). You will therefore have to install the appropriate server certificates and place them in a location (store), which can also be accessed by the service.

You can use the following steps: On the main page of the web service select **TestTimeStampService** and then click on the **Invoke** button (this is only possible with local access). This should produce a result similar to the one shown below:

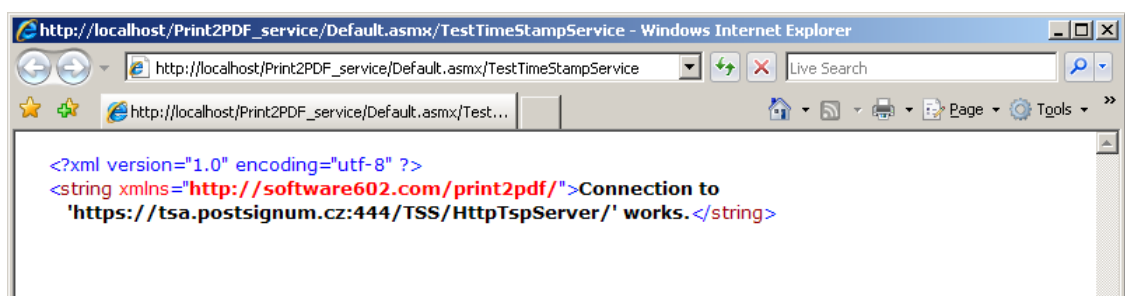


You will now have to add the root certificate of the TSA authority (in this case, it's PostSignum) into the system (not user) store used for the storage of trusted certificates. In order to do this, you will have to obtain the root certificate from the TSA provider (this is normally available as a free download). You will then have to install this certificate into the Local Computer store (it will be shown after **Show physical stores** has been checked):





The results of the TestTimeStampService operation should look similar to the one shown below:



There is a possibility that the TSA provider will require a certificate-based verification (instead of a username and password based verification).

In order to make sure the service can also identify itself using this certificate, the certificate must be imported using the [winhttpcertcfg.exe](http://www.microsoft.com/downloads/en/details.aspx?familyid=C42E27AC-3409-40E9-8667-C748E422833F&displaylang=en) (<http://www.microsoft.com/downloads/en/details.aspx?familyid=C42E27AC-3409-40E9-8667-C748E422833F&displaylang=en>).

The program must be launched with the following parameters:

```
winhttpcertcfg -i eport_cert_passwd 1234.pfx -c LOCAL_MACHINE\My -a  
IWAM_account -p Heslo
```

***To ensure proper functioning, the IIS account must also be used as the launch identity in the application pool.***

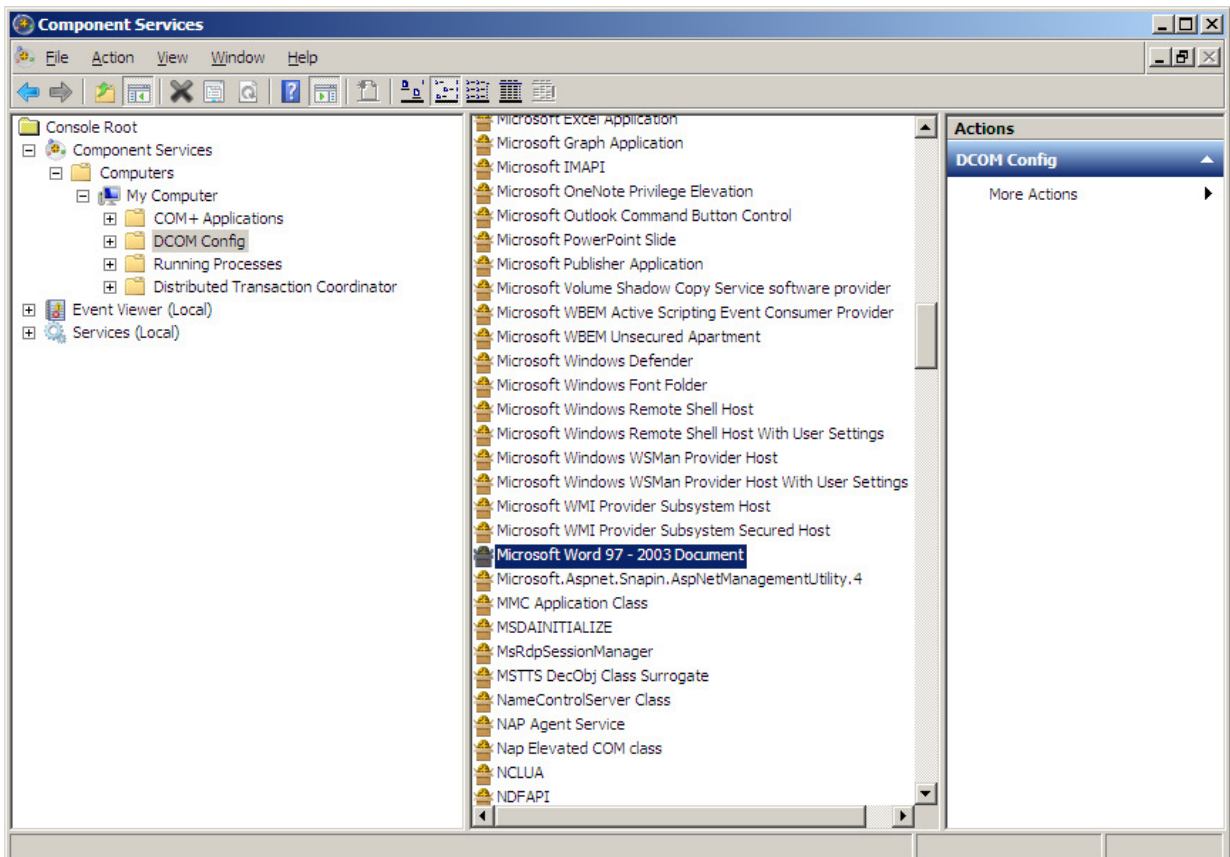
## 6.2. Converting Microsoft Office and Open Office Documents Using the Web Service

This feature may require additional permission settings for the web service for individual application COM objects. Not setting these permissions may cause problems when converting certain file formats. The resulting error will manifest itself in the web application as a VBS script error (although the specific script type can vary, depending on the type of file being converted).

### 6.2.1. Example of a DCOM Configuration

As an example, we have selected the configuration of a MS Word document object (these steps would be identical for other applications that are part of the MS Office suite and the examples shown below apply to Microsoft Server 200x and Windows Vista).

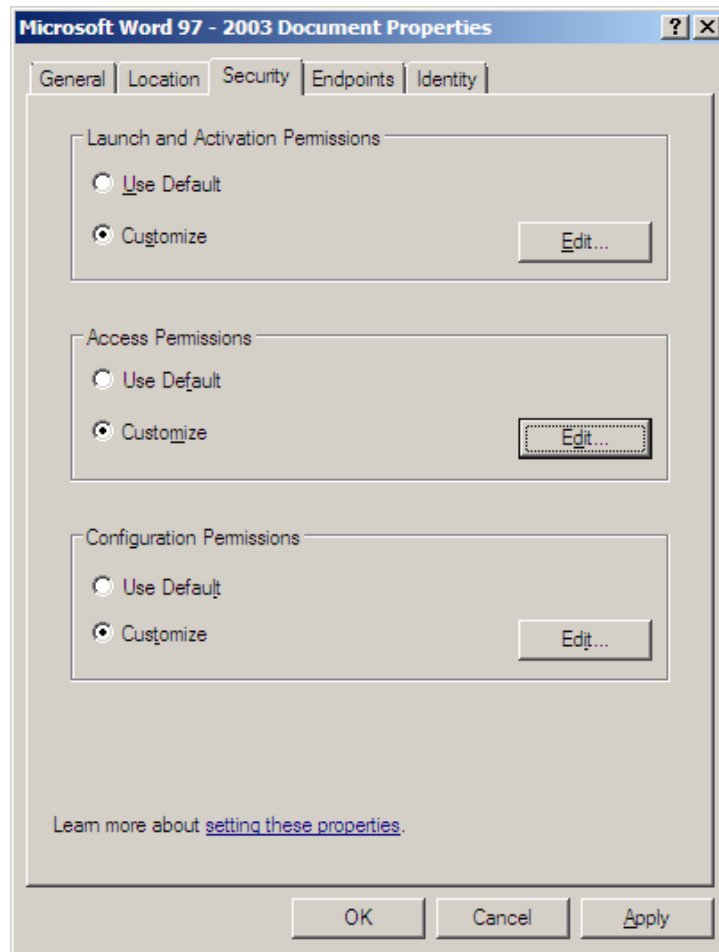
Run the dcomcnfg.exe tool (e.g. from the command line).



In **Component Services / My Computer / DCOM Config**, open the configuration of the application object (by clicking on **Properties** in the context menu).

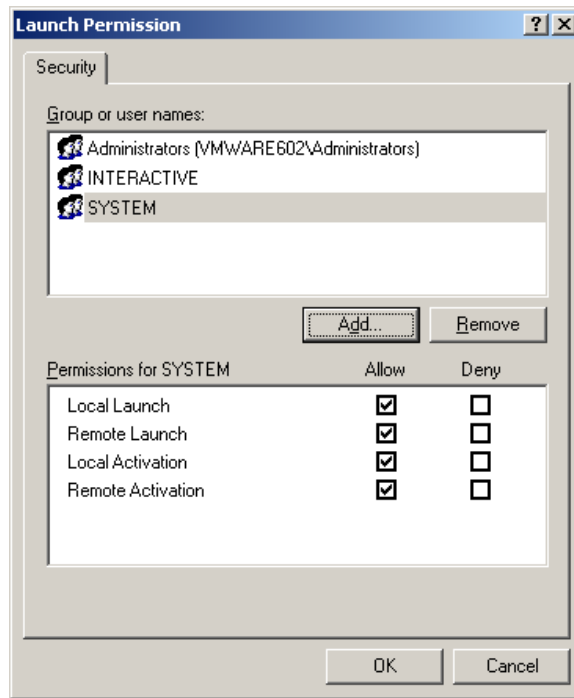
The DCOM object for the MS Word application is listed as **Microsoft Office Word 97-2003 Document**. If this object isn't listed, it was likely never registered. In such a situation, all you usually have to do is to open MS Word; then close it; and, then restart the server.

The settings that you will have to change for this conversion service normally include **Launch and Activation Permissions**, **Access Permissions** and sometimes also **Configuration Permissions**. These can be found under the **Security** tab of the **Properties** dialog box for the **Microsoft Word 97-2003 Document** object.

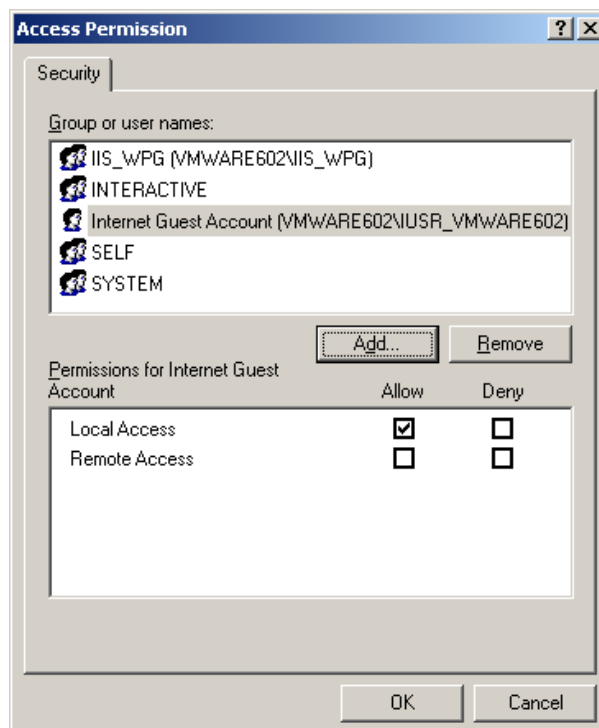


In the individual properties, you will have to add the 'Internet Guest Account' as one of the authorized users (this is normally listed as 'IUSR\_name\_of\_computer') and add the 'IIS Account' as one of the authorized users (this is normally 'IWAM\_name\_of\_computer'). Some server systems will also require that you add the 'IIS Administrations' group account (IISWG).

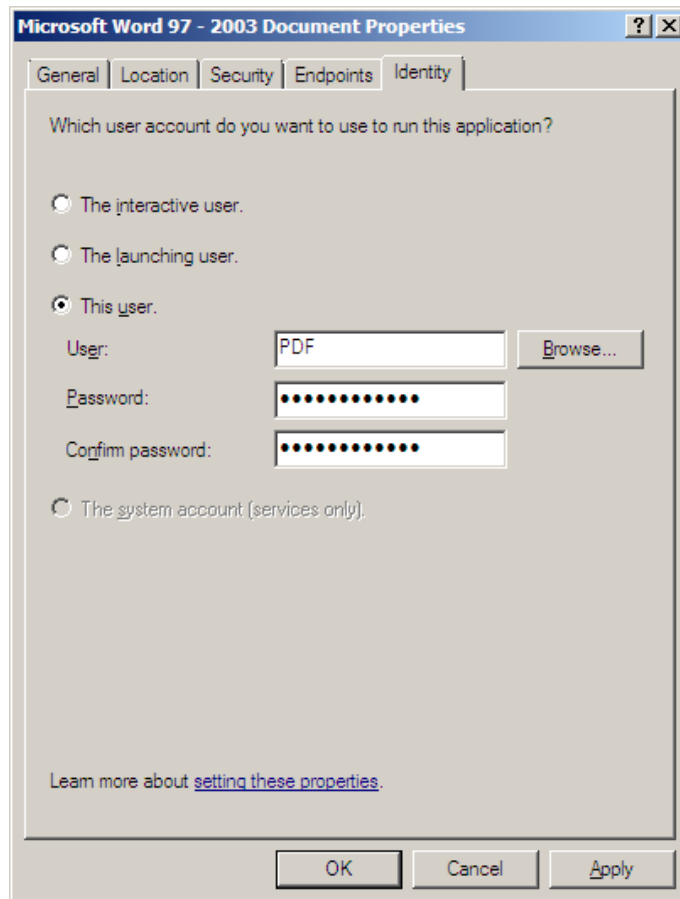
Also, add the 'Administrators' account to those included as listed users (if not already added) and make sure all permissions are checked as 'Allow' for this account:



Access permissions: Once again, you will need to add the same user accounts that you just did in the **Launch and Activation Permissions** section. Make sure local access is allowed for these accounts (only the 'SELF' and 'Administrators' accounts have remote access allowed).



The last setting you will have to change can be found under the **Identity** tab of the **Properties** dialog box, where you will have to check **The launching user.** as the user account the application will run under).



Another option is to select a specific user account (**This user.**) running the application (see below – configuration of the service for OpenOffice).

**Note: The Identity settings depend on the permission settings of the ASP.NET account. If the conversion is still not working after you have changed the settings (as instructed), you can try to set the same identity in the application pool used for the Print2PDF service and, possibly also, the web application from which the service is launched.**

You will normally need to change the setting for the converting of MS Word documents and OpenOffice documents (see below) and also possibly the settings for the converting of MS Excel, MS PowerPoint and MS Access files.

## 6.2.2. Converting OpenOffice Files Using the Web Service

In order to be able to convert OpenOffice files into the PDF format, you will need to change the OpenOffice configuration on the conversion server. You will have to modify the Setup.xcu configuration file (this is usually found in the share\registry\data\org\openoffice) folder.

To do this, complete the following steps:

Locate the following section in the Setup.xcu file:

```
<prop oor:name="ooSetupInstCompleted">  
<value>>false</value>
```

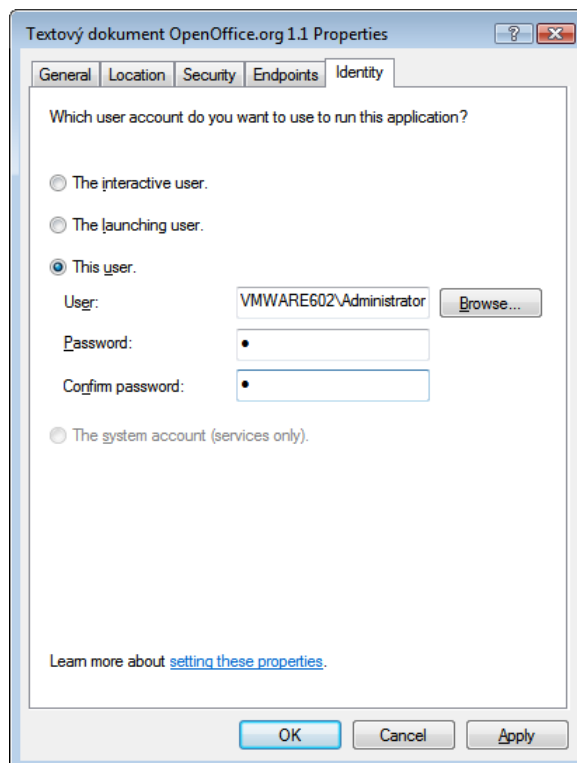
```
</prop>
<prop oor:name="ooSetupShowIntro">
  <value>true</value>
</prop>
```

Replace the above section with the following parameters:

```
<prop oor:name="ooSetupInstCompleted" oor:type="xs:boolean">
  <value>true</value>
</prop>
<prop oor:name="LicenseAcceptDate" oor:type="xs:string">
  <value>2006-07-25T17:34:04</value>
</prop>
<prop oor:name="FirstStartWizardCompleted" oor:type="xs:boolean">
  <value>true</value>
</prop>
```

As the next step, you will have to change the properties of the OpenOffice 1.1 Text Document DCOM object (using steps similar to those used in changing the settings for Microsoft Office application objects).

For the OpenOffice object, you will however need to set the **Identity** to **This user**, and enter the account of the launching user (this can be the 'Administrator' or you can set up a new account for the web application). This account is entered during the installation of the web application (see below) and it should be a part of the administrator accounts group.



The reason for the above difference is a different approach to access permissions in OpenOffice (classes and objects).

These settings can also be used for the configuration of MS Office applications.

## 7. Appendices

### 7.1. Appendix 1: WSDL Service Definition File

This file contains the formal parameters of the web service, per SOAP.

```
<?xml version="1.0" encoding="utf-8" ?>
_ <wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://software602.com/print2pdf/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
targetNamespace="http://software602.com/print2pdf/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
_ <wsdl:types>
_ <s:schema elementFormDefault="qualified"
targetNamespace="http://software602.com/print2pdf/">
_ <s:element name="ConvertFileAppend">
_ <s:complexType>
_ <s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="Input" type="s:base64Binary" />
<s:element minOccurs="0" maxOccurs="1" name="FileName" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="Profile" type="s:string" />
<s:element minOccurs="1" maxOccurs="1" name="AppendSessionID" type="s:int" />
</s:sequence>
</s:complexType>
</s:element>
_ <s:element name="ConvertFileAppendResponse">
_ <s:complexType>
_ <s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="ConvertFileAppendResult" type="s:int" />
<s:element minOccurs="0" maxOccurs="1" name="Output" type="s:base64Binary" />
<s:element minOccurs="0" maxOccurs="1" name="Report" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
_ <s:element name="ConvertFile">
_ <s:complexType>
_ <s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="Input" type="s:base64Binary" />
<s:element minOccurs="0" maxOccurs="1" name="FileName" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="Profile" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
_ <s:element name="ConvertFileResponse">
_ <s:complexType>
_ <s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="ConvertFileResult" type="s:int" />
<s:element minOccurs="0" maxOccurs="1" name="Output" type="s:base64Binary" />
<s:element minOccurs="0" maxOccurs="1" name="Report" type="s:string" />
</s:sequence>
</s:complexType>
```

```

</s:element>
_ <s:element name="GetProfileList">
<s:complexType />
</s:element>
_ <s:element name="GetProfileListResponse">
_ <s:complexType>
_ <s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="GetProfileListResult" type="s:int" />
<s:element minOccurs="0" maxOccurs="1" name="Profiles" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="Report" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
_ <s:element name="BeginAppendSession">
<s:complexType />
</s:element>
_ <s:element name="BeginAppendSessionResponse">
_ <s:complexType>
_ <s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="BeginAppendSessionResult" type="s:int"
/>
</s:sequence>
</s:complexType>
</s:element>
_ <s:element name="EndAppendSession">
_ <s:complexType>
_ <s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="SessionID" type="s:int" />
</s:sequence>
</s:complexType>
</s:element>
_ <s:element name="EndAppendSessionResponse">
<s:complexType />
</s:element>
_ <s:element name="GetTimeStamp">
_ <s:complexType>
_ <s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="Hash" type="s:string" />
<s:element minOccurs="1" maxOccurs="1" name="Flags" type="s:long" />
</s:sequence>
</s:complexType>
</s:element>
_ <s:element name="GetTimeStampResponse">
_ <s:complexType>
_ <s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="GetTimeStampResult" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="Error" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
_ <s:element name="TestTimeStampService">
<s:complexType />
</s:element>
_ <s:element name="TestTimeStampServiceResponse">
_ <s:complexType>

```

```

- <s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="TestTimeStampServiceResult"
type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="Test_ProfileList">
<s:complexType />
</s:element>
- <s:element name="Test_ProfileListResponse">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="Test_ProfileListResult" type="s:string"
/>
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="Test_ConvertFile_TXT">
<s:complexType />
</s:element>
- <s:element name="Test_ConvertFile_TXTResponse">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="Test_ConvertFile_TXTResult"
type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="Test_ConvertFile_DOC">
<s:complexType />
</s:element>
- <s:element name="Test_ConvertFile_DOCResponse">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="Test_ConvertFile_DOCResult"
type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
</s:schema>
</wsdl:types>
- <wsdl:message name="ConvertFileAppendSoapIn">
<wsdl:part name="parameters" element="tns:ConvertFileAppend" />
</wsdl:message>
- <wsdl:message name="ConvertFileAppendSoapOut">
<wsdl:part name="parameters" element="tns:ConvertFileAppendResponse" />
</wsdl:message>
- <wsdl:message name="ConvertFileSoapIn">
<wsdl:part name="parameters" element="tns:ConvertFile" />
</wsdl:message>
- <wsdl:message name="ConvertFileSoapOut">
<wsdl:part name="parameters" element="tns:ConvertFileResponse" />
</wsdl:message>
- <wsdl:message name="GetProfileListSoapIn">
<wsdl:part name="parameters" element="tns:GetProfileList" />
</wsdl:message>

```

```

- <wsdl:message name="GetProfileListSoapOut">
<wsdl:part name="parameters" element="tns:GetProfileListResponse" />
</wsdl:message>
- <wsdl:message name="BeginAppendSessionSoapIn">
<wsdl:part name="parameters" element="tns:BeginAppendSession" />
</wsdl:message>
- <wsdl:message name="BeginAppendSessionSoapOut">
<wsdl:part name="parameters" element="tns:BeginAppendSessionResponse" />
</wsdl:message>
- <wsdl:message name="EndAppendSessionSoapIn">
<wsdl:part name="parameters" element="tns:EndAppendSession" />
</wsdl:message>
- <wsdl:message name="EndAppendSessionSoapOut">
<wsdl:part name="parameters" element="tns:EndAppendSessionResponse" />
</wsdl:message>
- <wsdl:message name="GetTimeStampSoapIn">
<wsdl:part name="parameters" element="tns:GetTimeStamp" />
</wsdl:message>
- <wsdl:message name="GetTimeStampSoapOut">
<wsdl:part name="parameters" element="tns:GetTimeStampResponse" />
</wsdl:message>
- <wsdl:message name="TestTimeStampServiceSoapIn">
<wsdl:part name="parameters" element="tns:TestTimeStampService" />
</wsdl:message>
- <wsdl:message name="TestTimeStampServiceSoapOut">
<wsdl:part name="parameters" element="tns:TestTimeStampServiceResponse" />
</wsdl:message>
- <wsdl:message name="Test_ProfileListSoapIn">
<wsdl:part name="parameters" element="tns:Test_ProfileList" />
</wsdl:message>
- <wsdl:message name="Test_ProfileListSoapOut">
<wsdl:part name="parameters" element="tns:Test_ProfileListResponse" />
</wsdl:message>
- <wsdl:message name="Test_ConvertFile_TXTSoapIn">
<wsdl:part name="parameters" element="tns:Test_ConvertFile_TXT" />
</wsdl:message>
- <wsdl:message name="Test_ConvertFile_TXTSoapOut">
<wsdl:part name="parameters" element="tns:Test_ConvertFile_TXTResponse" />
</wsdl:message>
- <wsdl:message name="Test_ConvertFile_DOCSoapIn">
<wsdl:part name="parameters" element="tns:Test_ConvertFile_DOC" />
</wsdl:message>
- <wsdl:message name="Test_ConvertFile_DOCSoapOut">
<wsdl:part name="parameters" element="tns:Test_ConvertFile_DOCResponse" />
</wsdl:message>
- <wsdl:portType name="Print2PDF_WebServiceSoap">
- <wsdl:operation name="ConvertFileAppend">
<wsdl:input message="tns:ConvertFileAppendSoapIn" />
<wsdl:output message="tns:ConvertFileAppendSoapOut" />
</wsdl:operation>
- <wsdl:operation name="ConvertFile">
<wsdl:input message="tns:ConvertFileSoapIn" />
<wsdl:output message="tns:ConvertFileSoapOut" />
</wsdl:operation>
- <wsdl:operation name="GetProfileList">

```

```

<wsdl:input message="tns:GetProfileListSoapIn" />
<wsdl:output message="tns:GetProfileListSoapOut" />
</wsdl:operation>
_ <wsdl:operation name="BeginAppendSession">
<wsdl:input message="tns:BeginAppendSessionSoapIn" />
<wsdl:output message="tns:BeginAppendSessionSoapOut" />
</wsdl:operation>
_ <wsdl:operation name="EndAppendSession">
<wsdl:input message="tns:EndAppendSessionSoapIn" />
<wsdl:output message="tns:EndAppendSessionSoapOut" />
</wsdl:operation>
_ <wsdl:operation name="GetTimeStamp">
<wsdl:input message="tns:GetTimeStampSoapIn" />
<wsdl:output message="tns:GetTimeStampSoapOut" />
</wsdl:operation>
_ <wsdl:operation name="TestTimeStampService">
<wsdl:input message="tns:TestTimeStampServiceSoapIn" />
<wsdl:output message="tns:TestTimeStampServiceSoapOut" />
</wsdl:operation>
_ <wsdl:operation name="Test_ProfileList">
<wsdl:input message="tns:Test_ProfileListSoapIn" />
<wsdl:output message="tns:Test_ProfileListSoapOut" />
</wsdl:operation>
_ <wsdl:operation name="Test_ConvertFile_TXT">
<wsdl:input message="tns:Test_ConvertFile_TXTSoapIn" />
<wsdl:output message="tns:Test_ConvertFile_TXTSoapOut" />
</wsdl:operation>
_ <wsdl:operation name="Test_ConvertFile_DOC">
<wsdl:input message="tns:Test_ConvertFile_DOCSoapIn" />
<wsdl:output message="tns:Test_ConvertFile_DOCSoapOut" />
</wsdl:operation>
</wsdl:portType>
_ <wsdl:binding name="Print2PDF_WebServiceSoap" type="tns:Print2PDF_WebServiceSoap">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
_ <wsdl:operation name="ConvertFileAppend">
<soap:operation soapAction="http://software602.com/print2pdf/ConvertFileAppend"
style="document" />
_ <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
_ <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
_ <wsdl:operation name="ConvertFile">
<soap:operation soapAction="http://software602.com/print2pdf/ConvertFile"
style="document" />
_ <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
_ <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
_ <wsdl:operation name="GetProfileList">

```

```
<soap:operation soapAction="http://software602.com/print2pdf/GetProfileList"
style="document" />
_ <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
_ <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
_ <wsdl:operation name="BeginAppendSession">
<soap:operation soapAction="http://software602.com/print2pdf/BeginAppendSession"
style="document" />
_ <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
_ <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
_ <wsdl:operation name="EndAppendSession">
<soap:operation soapAction="http://software602.com/print2pdf/EndAppendSession"
style="document" />
_ <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
_ <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
_ <wsdl:operation name="GetTimeStamp">
<soap:operation soapAction="http://software602.com/print2pdf/GetTimeStamp"
style="document" />
_ <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
_ <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
_ <wsdl:operation name="TestTimeStampService">
<soap:operation soapAction="http://software602.com/print2pdf/TestTimeStampService"
style="document" />
_ <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
_ <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
_ <wsdl:operation name="Test_ProfileList">
<soap:operation soapAction="http://software602.com/print2pdf/Test_ProfileList"
style="document" />
_ <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
_ <wsdl:output>
```

```
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
_ <wsdl:operation name="Test_ConvertFile_TXT">
<soap:operation soapAction="http://software602.com/print2pdf/Test_ConvertFile_TXT"
style="document" />
_ <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
_ <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
_ <wsdl:operation name="Test_ConvertFile_DOC">
<soap:operation soapAction="http://software602.com/print2pdf/Test_ConvertFile_DOC"
style="document" />
_ <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
_ <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
_ <wsdl:binding name="Print2PDF_WebServiceSoap12"
type="tns:Print2PDF_WebServiceSoap">
<soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
_ <wsdl:operation name="ConvertFileAppend">
<soap12:operation soapAction="http://software602.com/print2pdf/ConvertFileAppend"
style="document" />
_ <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
_ <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
_ <wsdl:operation name="ConvertFile">
<soap12:operation soapAction="http://software602.com/print2pdf/ConvertFile"
style="document" />
_ <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
_ <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
_ <wsdl:operation name="GetProfileList">
<soap12:operation soapAction="http://software602.com/print2pdf/GetProfileList"
style="document" />
_ <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
_ <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
```

```
</wsdl:operation>
_ <wsdl:operation name="BeginAppendSession">
<soap12:operation soapAction="http://software602.com/print2pdf/BeginAppendSession"
style="document" />
_ <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
_ <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
_ <wsdl:operation name="EndAppendSession">
<soap12:operation soapAction="http://software602.com/print2pdf/EndAppendSession"
style="document" />
_ <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
_ <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
_ <wsdl:operation name="GetTimeStamp">
<soap12:operation soapAction="http://software602.com/print2pdf/GetTimeStamp"
style="document" />
_ <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
_ <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
_ <wsdl:operation name="TestTimeStampService">
<soap12:operation soapAction="http://software602.com/print2pdf/TestTimeStampService"
style="document" />
_ <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
_ <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
_ <wsdl:operation name="Test_ProfileList">
<soap12:operation soapAction="http://software602.com/print2pdf/Test_ProfileList"
style="document" />
_ <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
_ <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
_ <wsdl:operation name="Test_ConvertFile_TXT">
<soap12:operation soapAction="http://software602.com/print2pdf/Test_ConvertFile_TXT"
style="document" />
_ <wsdl:input>
<soap12:body use="literal" />
```

```
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="Test_ConvertFile_DOC">
<soap12:operation soapAction="http://software602.com/print2pdf/Test_ConvertFile_DOC"
style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
- <wsdl:service name="Print2PDF_WebService">
- <wsdl:port name="Print2PDF_WebServiceSoap" binding="tns:Print2PDF_WebServiceSoap">
<soap:address location="http://localhost/WebService_x64/Default.asmx" />
</wsdl:port>
- <wsdl:port name="Print2PDF_WebServiceSoap12"
binding="tns:Print2PDF_WebServiceSoap12">
<soap12:address location="http://localhost/WebService_x64/Default.asmx" />
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

## 7.2. Appendix 2: Error Codes for the ConvertFile Function

<b>Code</b>	<b>Result</b>
0	Convert OK...
1	Unknown error
2	Unknown error during initialization
3	Unknown error during printing
4	Operation cancelled
5	Operation cancelled during initialization
6	Operation cancelled during printing
7	Conversion error
8	Conversion error during initialization
9	Conversion error during printing
10	Unable to run Word macro
11	Operation timed out during file conversion
12	Operation timed out during initialization
13	There is no application associated with the given file extension
14	Unable to open host application
15	Unable to open input document
16	Not enough memory to continue
17	Operation timed out during page conversion
18	The document is password protected
19	Host application terminated
20	Host application terminated during initialization
21	Host application terminated during printing
22	Operation timed out while waiting in the queue
23	Unable to change application setting
24	Unable to change document setting
25	Unable to set default printer
26	Unable to create temporary data
27	Unable to connect to Word macro

<b>Code</b>	<b>Result</b>
28	Error during Word macro execution
29	Unable to find Word macro
30	Operation timed out during Word macro execution
31	Unknown error during Word macro execution
32	Unsupported file name extension
33	Invalid signature (DigitalSignature returned 0 (VARIANT_BOOL false))
34	The license key that is provided does not enable digital signatures
35	The document security settings do not allow the user to modify the document
36	One of the arguments is invalid or the certificate does not allow signing documents
37	Unspecified DigitalSignature error
38	Conversion script for this file type not found
39	Error reading conversion script or unsupported encoding
40	ShellExecute The operating system is out of memory or resources
42	The specified file was not found.
43	The specified path was not found.
45	The operating system denied access to the specified file.
48	There was not enough memory to complete the operation.
51	The .exe file is invalid (non-Microsoft Win32 .exe or error in .exe image).
66	A sharing violation occurred.
67	The file name association is incomplete or invalid.
68	The DDE transaction could not be completed because the request timed out.
69	The DDE transaction failed.
70	The Dynamic Data Exchange (DDE) transaction could not be completed because other DDE transactions were being processed.
72	The specified dynamic-link library (DLL) was not found.
73	ShellExecute return error codes <= 32
75	Conversion script failed, try next script in list
76	Extension not listed in AllowedFileTypes
100	Could not connect Print2PDF windows service.
101	Invalid command %s



<b>Code</b>	<b>Result</b>
102	Some parameters are missing
103	Exported function is missing in Print602.dll