



dox42 Server

V3.5.1-4.1

Documentation

Summary

This document explains the service interface as well as the configuration for the dox42 server.

Content

Summary.....	2
Content.....	2
Document details	3
Service interface	4
SOAP Interface.....	4
REST Interface.....	5
Service-Header.....	6
Azure Active Directory Header.....	6
OutputActions.....	7
SaveAction.....	8
SharepointAction.....	9
Set special SharePoint and Managed Metadata Fields.....	11
EmailAction	12
ReturnAction.....	15
PrintAction.....	16
dox42LIVEAction.....	16
Configuration parameters	17
CORS Configuration.....	19
Azure Active Directory Access	20
Custom Error Messages Sample.....	20
Hyphenation Dictionaries Sample.....	20
Maximum size of request or time-out.....	20
License Activation.....	21
Installation.....	22
Upgrade.....	24
Support.....	24

Document details

Version: dox42 server V3.5.1-4.0

Author: Christian Bauer

Date: 08 January 2019

Service interface

The dox42 server is accessible via a SOAP and a REST Webservice interface.

Operation: GenerateDocument/GenerateSlides

The operation can be operated with the **following parameters**:

- › **DocTemplate** (String)
path to the document template (e.g. C:\doctemplates\solar_subsidy.docx, C:\doctemplates\salesreport.pptx)
- › **InputParams** (Key-Value String/String)
list of the input parameters and values:
 - › **ParamName**: name of the parameter (not case-sensitive)
 - › **Value**: value of the parameter
- › **PostGenActions**
list of PostGenerateActions to process after generating the document:
 - › **ActionName**: name of the action (e.g. SaveAction), not case-sensitive
 - › **Params**: parameter of the action (like InputParams)

Operation: GenerateSpreadSheet

The operation can be operated with the **following parameters**:

- › **DataMap** (String)
path to the data map (e.g. C:\doctemplates\salesreport.dm)
- › **InputParams** (Key-Value String/String)
list of the input parameters and values:
 - › **ParamName**: name of the parameter (not case-sensitive)
 - › **Value**: value of the parameter
- › **PostGenActions**
list of PostGenerateActions to process after generating the document:
 - › **ActionName**: name of the action (e.g. SaveAction), not case-sensitive
 - › **Params**: parameter of the action (like InputParams)

The paramameters of all individual OutputActions are described in this document.

SOAP Interface

URL: <http://www.YourDomain.com/dox42service.asmx>

WSDL: <http://www.YourDomain.com/dox42service.asmx?WSDL>

REST Interface

URL: <http://www.YourDomain.com/dox42RestService.ashx>

The parameters and output actions for the REST and the SOAP interface are identical. The REST Interface uses the following URL-parameter syntax. Since version 3.6.3 you can also pass all parameter via POST:

Parametername	Erfordernis	Auswirkung
Operation	mandatory GenerateDocument GenerateSpreadSheet	The operation to call
DocTemplate	mandatory if Operation=GenerateDocument	Path of the document template
DataMap	mandatory if Operation= GenerateSpreadSheet	Path of the data map
InputParam. <ParameterName>	For each input parameter	Values for input parameters, e.g.: InputParam.CustomerID=42
<OutputAction>. <ParameterName>	For each parameter of each OutputAction	Values for OutputAction- Parameters, z.B.: ReturnAction.Format=pdf
ReturnAction.disp	optional Inline attachment	Controls the how the web browser handles the returned document
ReturnAction.Mode	optional File (Standard) Text	Use Text to render the content of the generated document directly in the browser as text. This text may be HTML.
ReturnAction.fileName	optional, only if ReturnAction.disp = attachment	Name of the returned document
RedirectURL	Optional, only without ReturnAction	Redirect to this page after generation

Caution: Use URL-Encoding! (http://www.w3schools.com/tags/ref_urlencode.asp):

„Customer.Name=Müller“ => „Customer.Name=M%C3%BCller “

Sample for a REST call:

[http://www.YourDomain.com/dox42RestService.ashx?
Operation=GenerateDocument&
DocTemplate=%7e%5cTemplates%5cSales_Report%5cSales_Report.doc &
InputParam.SelectEmployee=5&InputParam.SelectYear=2012&
ReturnAction.format=pdf&ReturnAction.fileName=SalesReport.pdf&ReturnAction.disp=attachment](http://www.YourDomain.com/dox42RestService.ashx?Operation=GenerateDocument&DocTemplate=%7e%5cTemplates%5cSales_Report%5cSales_Report.doc&InputParam.SelectEmployee=5&InputParam.SelectYear=2012&ReturnAction.format=pdf&ReturnAction.fileName=SalesReport.pdf&ReturnAction.disp=attachment)

HTTP Request-Body

In case you need to transfer lots of data to the dox42 Server you better do that as POST parameters or in the Request-Body (since version 4.0). To extract the Request-Body you need to define an InputParameter of the name „RequestBody“ in your DataMap. The data from the HTTP Request-Body will be stored to this InputParameter and can be processed using a XML/JSON data source for example.

Service-Header

A service header secures the dox42 SOAP service. This header contains the user name and the password. Use web.config to configure the utilisation of the service header.

Azure Active Directory Header

The dox42 Service can process an Authorization Header, containing an Azure Active Directory Token (since version 4.0).

Javascript Example:

```
xhttp.setRequestHeader("Authorization", 'Bearer ' + token);
```

OutputActions

The product **dox42** server contains the following OutputActions:

- › **SaveAction:** saves the generated document to a file folder
- › **SharepointAction:** saves the generated document to a Sharepoint document library
- › **EmailAction:** sends the generated document via email
- › **ReturnAction:** returns the generated document via the WebServices interface
- › **PrintAction:** print generated document
- › **CustomOutputAction:** You can implement and use your own dox42 OutputAction. Custom Output Actions have to be registered in the web.config (analog Custom Data Sources) and use the following interface:
dox42CustomOutputActionKit. [ICustomOutputAction](#)

In the following, the parameters of the OutputActions are described. None of the parameter names are case-sensitive. All parameter names may contain data fields (for exceptions see table). Example:
FileName = "Application_<%Citizen.Name%>.pdf"

SaveAction

Parameter name	Requirement	Effect
FileName	mandatory	<p>Defines the name of the document, which it will be saved as. e.g. C:\Documents\SalesReport.pdf</p> <p>The extension will define the file format (.docx, .xlsx, .pptx, .pdf, .html, ...). See all output formats below.</p>
SpecialFormat (dox42ForDocuments only)	optional	<p>When generating a PDF, the format may be specified: PDFA1b and PDFForm. For PDFForm Form fields are preserved in the PDF.</p> <p>For HTML, you may use these option: HTMLCode which will cause the text of the dox42 template to be interpreted in HTML code.</p> <p>HTMLZIP The HTML -file along with the contained images will be packaged in a ZIP-file. (since version 3.6.3).</p> <p>For text files (*.txt) you can set an alternate extension. That is useful for generating meta data files for archiving-systems. Example: FileName = Order_<%Cust.OderNo%>.txt SpecialFormat = jpl Result: Order_42.jpl (since version 3.5)</p>
Sign	optional	<p>Value: true</p> <p>The format PDF may bear a digital signature (dox42ForDocuments only). Certificate and corresponding password have to be defined in the web.config.</p>

Possible Output Formats

Word: doc, docx, docm, dot, dotx, pdf, html, txt, jpg, png, tiff, bmp, rtf, odt
 Excel: xls, xlsx, xlsxm, xlt, xltx, xltm, xlsb, txt, csv, html, pdf, xml, tiff, ods
 PowerPoint: pptx, pps, pdf, tiff

SharepointAction

Parameter name	Requirement	Effect
Site	mandatory	The Sharepoint site e.g. http://mysite.com
Library	mandatory	The Sharepoint document library root folder name
FileName	mandatory	Defines the name of the document, which it will be saved as. e.g. SalesReport.pdf The extension will define the file format (.docx, .xlsx, .pptx, .pdf, .html, ...).
Folder	optional	Saves the document to the specified folder.
SpecialFormat (dox42ForDocuments only)	optional	When generating a PDF, the format may be specified: PDFA1b and PDFForm. For PDFForm Form fields are preserved in the PDF. For HTML, you may use the option: HTMLCode which will cause the text of the generated document to be interpreted in HTML code.
Sign	optional	Value: true The format PDF may bear a digital signature (dox42ForDocuments only). Certificate and corresponding password have to be defined in the web.config.
Field1.Name Field1.Value	optional	Fields allow you to write values to columns in the Sharepoint document library. You can create multiple Fields: Field1, Field2, ... Use the SharePoint internal field name to identify the field. Tipp: You may find out the internal field name by opening the list in the dox42 SharePoint data source and check "Use Internal Field Names".
ContentType	optional	Set this Content Type for the item

DocumentDataField	optional	<p>If this field is assigned, the generated document will not be saved in the document library, but the document in the stated dox42 Datafield will be loaded (e.g. Data.MyDocument) and saved in the Sharepoint document library.</p> <p>The DocumentDataField can contain a Path, an URL or a binary Document.</p>
CheckInMessage	optional	<p>If this Field is assigned, the server will check in the document after saving with the stated message. If a document of the same title is already existing this document will be checked out automatically</p>
Username	optional	<p>If not provided, the service account of the dox42 Server will be used for SharePoint authentication.</p>
Password	optional	
SharePoint_online	optional	<p>true/false Set this flag for SharePoint online (Office 365). Alternative to Azure_AD</p>
SharePoint_FBA	optional	<p>true/false Set this flag for SharePoint sites with forms based authentication.</p>
Azure_AD	optional	<p>true/false The Request must include an Authorization Header containing Azure Active Directory Token - alternative to SharePoint_online. (Since version 4.0).</p>
retry_count	optional	<p>A number ≥ 0 Defines that the SharePointAction should retry to save the document in case of error (e.g. timeout) and specifies the number of retries. Default = 0 (no retry). (Since version 4.1.3).</p>

Set special SharePoint and Managed Metadata Fields

For a choice field simply set the value.

To set a multi value choice field use “;#” to separate the values, e.g.: “red;#green;#yellow”.

For a Lookup/Person/Group-field set the ID.

To set a multi value Lookup/Person/Group-field, you need to set the ID and the value like „ID;#Value“, e.g.: „3;#Hawaii;#9;#Vienna;#13;#Melbourne“. Actually only the IDs are used to set the field, the values are ignored (but may not be omitted).

EmailAction

Parameter name	Requirement	Effect
Mode	optional Possible Values: TxtEMail HtmlEMail Attachment (default)	Allows to configure whether the generated document should be used as the body of a txt- or html-mail or attached to the email. Use TxtEMail and HtmlEMail only with GenerateDocument.
FileName	mandatory if mode = Attachment, otherwise ignored	Name of the attachment The extension will define the file format (.docx, .xlsx, .pptx, .pdf, .html, ...).
SpecialFormat (dox42ForDocuments only)	optional	When generating a PDF, the format may be specified: PDFA1b and PDFForm. For PDFForm Form fields are preserved in the PDF. For HTML, you may use the option: HTMLCode which will cause the text of the generated document to be interpreted in HTML code.
Sign	optional	Value: true The format PDF may bear a digital signature (dox42ForDocuments only). Certificate and corresponding password have to be defined in the web.config.
Receiver	mandatory	Receivers email addresses Separate multiple addresses with a comma.
ReceiverCC	optional	Carbon copy receivers
ReceiverBCC	optional	Blind carbon copy receivers
Sender	optional	Sender's email address If no email address is specified, the address stated in the web.config will be used.

Parameter name	Requirement	Effect
MailServerUsername	optional	May be specified along with Sender if a different login for the mail server is required. Since Version 3.6.2.4
SenderPassWord	mandatory if Sender or MailServerUsername is specified, else: omit	Sender's password for his email account or MailServerUsername on the email server, configured in the web.config.
Header	mandatory	Email subject
Body	optional	Email text (mode = Attachment only)
HtmlBody	optional	Email text as HTML (mode = Attachment only)
BodyTemplate	optional	A dox42 Word Template to generate the email body. The template must use the same Datamap as the attachment. (mode = Attachment only) Since Version 3.5.1
StaticAttachment1.Path	optional (or use Base64) e.g. C:\Images\MyImage.jpg	Allows sending for static attachments. It is possible to send any number of attachments, like: StaticAttachment2... StaticAttachment3... You may specify either the path of a static file, or pass the attachment base64 encoded.
StaticAttachment1.Base64	optional (or use Path) Base64 encoded attachment	
StaticAttachment1.Name	mandatory if Base64 is used e.g. MyImage.jpg	
MailServer	optional	Use this parameter if you want to use a different mail server than configured in your web.config.
MailServerPort	Optional Since version 3.4.3.10	Change the SMTP Mail Server Port
SSL	Optional Since version 3.5	true or false
SaveMSGFile	Optional Since version 3.5.0.19	File path to save the sent mail as msg-file, e.g. C:\Windows\Temp\Order_<%Customer.ID%>.msg

Parameter name	Requirement	Effect
SendMode	Optional Since Version 3.6	send or save send: Default, send mail immediately save: Don't send mail, but save the mail as draft to the path defined in SaveMSGFile. The parameter SaveMSGFile must be specified. return: Don't send mail, but return the mail as draft in the ReturnMessage. Since Version 3.6.3.

To use an alias with an email address please use this syntax (since 3.6.2.5):

MyCompany Sales {sales@mycompany.com}

MyCompany Sales <sales@mycompany.com> , would work as well, but might cause an Error if used with the REST Service.

ReturnAction

Parameter name	Requirement	Effect
Format	mandatory	File format (.docx, .xlsx, .pptx, .pdf, .html, ...).
Mode	optional Possible Values: File (Standard) Text	Defines whether the generated Document should be returned as File (Mode=File) in ServiceResult.GeneratedDocs or as text (Mode=Text) in ServiceResult.ResultMessage.
SpecialFormat (dox42ForDocuments only)	optional	When generating a PDF, the format may be specified: PDFA1b and PDFForm. For PDFForm Form fields are preserved in the PDF. For HTML, you may use the option: HTMLCode which will cause the text of the generated document to be interpreted in HTML code.
Sign	optional	Value: true The format PDF may bear a digital signature (dox42ForDocuments only). Certificate and corresponding password have to be defined in the web.config.

PrintAction

Parameter name	Requirement	Effect
PrinterName	optional	The used printer
Collate	optional (true/false)	Sort?
Copies	optional	Amount of copies
Duplex	Optional (Vertical, Horizontal, Simplex)	Duplexprint
FromPage	optional	Print from pagenumber ...
ToPage	optional	... to pagenumber

dox42LIVEAction

The dox42LIVEAction uses an output configuration from a dox42LIVE.config file to execute output actions. Please see the dox42 LIVE documentation to learn how to define your dox42LIVE.config. To use a dox42LIVEAction from the dox42 SOAP or REST Service the dox42LIVE.config file needs to be stored in the same directory as dox42RestService.ashx and dox42Service.asmx , typically:

C:\dox42Server.

Since version 4.1.3.

Parameter name	Requirement	Effect
LiveActionName	mandatory	The LiveActionName as defined in the dox42LIVE.config file.

Configuration parameters

The following parameters can be configured for the **dox42** server in the AppSettings of the web.config.

- › **LicenseFilePath:** Indicates the path to the location of the license file. In case no license file exists, the **dox42** server will contact the **dox42** license server in order to verify the license key. If successful, the server will write the license file to the location under the path indicated in the configuration.
- › **LicenseKey:** license key of the **dox42** server
- › **MailServer:** email server for sending emails (optional)
- › **MailServerPort:** email server port (optional)
- › **dox42ServerMail:** sender's default address of the **dox42** service (optional)
- › **MailServerPassWord:** password for the **dox42** email server (optional)
- › **SendErrorsToMail:** standard email address for error messages (optional)
- › **Certificate:** certificate, used for signing (optional)
- › **CertificatePassWord:** certificate password (optional)
- › **CheckServiceHeader (true/false):** Indicates, whether the service header is to be analysed or not. If "false", no service header needs to be included.
- › **UserPermissionODBCConnectionString:** ODBC connection string to a data base, which contains user permissions
(mandatory if CheckServiceHeader = true)
- › **CheckUserPermissionSQL:** The select statement, which is executed on the UserPermissionODBCConnectionString data base. The statement has to return 1 or "true" in the case of a successfully authorised user. Within the statement, the %username% and %password% tags will be replaced with the values, included in the header.
(mandatory if CheckServiceHeader = true)
- › **ThrowExceptionOnError (true/false):**
true: The service returns an exception on error.
false (or not present): An error will be returned as ResultMessage.
This configuration parameter is available since build 3.3.0.8.
- › **CustomErrorMessageFile:**
Path of a XML Files containing Custom Error Messages (optional). Custom Error Messages allow for replacing Error Messages on the basis of a text search. Please find a sample later in the chapter.
- › **HyphenationDictionariesFile:**
Path of a XML Files containing containing Hyphenation Dictionaries (optional). Please find a sample later in the chapter. (Since version 3.4.3.10)

- › **CheckTrustedTemplateLocations (true/false):**
 - true:** The path/link of the template or data map must start with one of the paths/links listed in the section „trustedTemplateLocations“. If not, the call is cancelled.
 - false (or not present):** No check is performed.
 - CAUTION: if you switch off this check, everybody who is allowed to call your dox42 service can generate templates/datamaps from any location including unknown URLs on your dox42 server. We strongly recommend to make sure this check is always switched on for security reasons!**
 - This configuration parameter is available since build 3.3.4.
 - List the trusted template locations in the section „trustedTemplateLocations“.
- › **TempFolder:**
 - Folder to be used by the dox42 Server to store temporary files. If not configured the Windows standard Temp Folder is used.
 - This configuration parameter is available since version 3.5
- › **RESTInterfaceReturnsFullErrorInfoTempFolder:**
 - If true more detailed error information is returned for the dox42 REST call.
 - This configuration parameter is available since version 3.6.1.
- › **RESTInterfaceReturnsHTTPError500 (true/false):**
 - if true the http Statuscode 500 will be set on error for the dox42 REST call.
 - This configuration parameter is available since version 4.0.1
- › **SOAPServiceReturnsFullErrorInfo (true/false):**
 - If true more detailed error information is returned for the dox42 SOAP call.
 - This configuration parameter is available since version 4.1.1.0

Example

<appSettings>

```

<add key="LicenseFilePath" value="c:\dox42\dox42WebService\License" />
<add key="LicenseKey" value="..." />
<add key="MailServer" value="smtp.1und1.de" />
<add key="dox42ServerMail" value="testserver@dox42.com" />
<add key="MailServerPassWord" value="..." />
<add key="SendErrorsToMail" value="monitoring@dox42.com" />
<add key="Certificate" value="c:\dox42\dox42WebService\Certs\Certificate.pfx" />
<add key="CertificatePassWord" value="..." />
<add key="CheckServiceHeader" value="true" />
<add key="UserPermissionODBCConnectionString"
    value="Driver={SQL Server Native Client 10.0};Server=...;Database=...;Uid=...;Pwd=...;" />
<add key="CheckUserPermissionSQL"
    value="SELECT COUNT(*) FROM Users where Username =
    '%username%' and Password = '%password%' " />
<add key="ThrowExceptionOnError" value="false" />
<add key="CustomErrorMessageFile" value="C:\dox42Server\CustomErrors.xml" />
<add key="CheckTrustedTemplateLocations" value="true" />
<add key="TempFolder" value="C:\dox42Server\WorkingDir" />
<add key="RESTInterfaceReturnsFullErrorInfo" value="false" />
<add key="RESTInterfaceReturnsHTTPError500" value="false" />
<add key="SOAPServiceReturnsFullErrorInfo" value="false" />

```

</appSettings>

Custom data sources will be registered in a separate section under the name “customDataSources”.

```
<customDataSources>
  <add key="CSVDataEngine.CSVDataSourceParser"
    value="CSV Datenquelle;c:\dox42\dox42WebService\bin\CSVDataEngine.dll;" />
</customDataSources>
```

Custom Output Actions Sources are registered in a specified section called “customOutputActions”.

```
<customOutputActions>
  <add key="MyCustomOutputActions.MyOutputAction"
    value=" MyOutputAction; c:\dox42\dox42WebService\bin\MyCustomOutputActions.dll;" />
</customOutputActions>
```

Trusted template locations are registered in a specified section called “trustedTemplateLocations”.

```
<trustedTemplateLocations>
  <add key="local" value="C:\dox42Server\templates" />
  <add key="sharepoint" value="http://www.mysharepoint.mycompany.com/dox42Templates" />
</trustedTemplateLocations>
```

Configure log settings in the file NLog.config. For a complete documentation of the logging framework NLog see www.nlog-project.org.

CORS Configuration

If a web-application calls the dox42 Server e.g. via Javascript XMLHttpRequest under a different URL than the application itself, you might get a Cross-Origin Resource Sharing (CORS) error.

Example:

Calling web-application: <https://yourcompany.sharepoint.com/yoursite>

dox42 Server: <https://dox42server.yourcompany.com>

To enable this scenario you could add a customHeaders section to your web.config, example:

```
<system.webServer>
  <httpProtocol>
    <customHeaders>
      <clear />
      <add name="Access-Control-Allow-Origin" value="*" />
      <add name="Access-Control-Allow-Headers"
        value="Content-Type, Accept, Authorization" />
      <add name="Access-Control-Allow-Methods" value="GET,POST,OPTIONS" />
      <add name="Access-Control-Expose-Headers"
        value="Content-Type, Content-Disposition, Accept" />
    </customHeaders>
  </httpProtocol>
</system.webServer>
```

See also:

<https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>

<http://www.w3.org/TR/cors/>

Azure Active Directory Access

This is necessary to store dox42 Templates and Data-Maps on SharePoint Online.

ConfigSection:

```
<configSections>
    . . .
    <section name ="azureAD" type ="dox42.Core.Utils.AzureADSectionHandler,dox42Core"/>
</configSections>
```

Azure AD Section:

```
<azureAD>
  <add resource365="https://mytenant.sharepoint.com"
        appID="..."
        tenant="..."
        clientKey="*****" />
</azureAD>
```

Custom Error Messages Sample

```
<?xml version="1.0" encoding="utf-8"?>
<CustomErrorMessagesDS xmlns="http://dox42.com/CustomErrorMessagesDS.xsd">
  <CustomErrorMessage>
    <Test2SearchInExc>Error uploading file http://mysharepoint</Test2SearchInExc>
    <NewMessage>
SharePoint-farm not responding, please contact your administrator</NewMessage>
  </CustomErrorMessage>
</CustomErrorMessagesDS>
```

In case you edit this file using a text editor, please be sure to save in UTF-8 encoding.

Hyphenation Dictionaries Sample

```
<?xml version="1.0" standalone="yes"?>
<HyphenationDictionary xmlns="http://www.dox42.com/HyphenationDictionary.xsd">
  <Dictionary>
    <Culture>de-CH</Culture>
    <DictionaryPath>C:\dox42Server\hyph_de_CH.dic</DictionaryPath>
  </Dictionary>
</HyphenationDictionary>
```

To use automatic hyphenation with the dox42 Word Add-In save such a file named HyphenationDictionary.config to C:\Users\<<USER>>\AppData\Local\dox42V1

Maximum size of request or time-out

Should you be sending large of requests to dox42, return large files or start a bulk letter, it might become necessary to increase request size or time-out. (The default ASP.NET time-out is 90 sec. After that, a longer call will be aborted by IIS.) Configure this in the web.config as well.

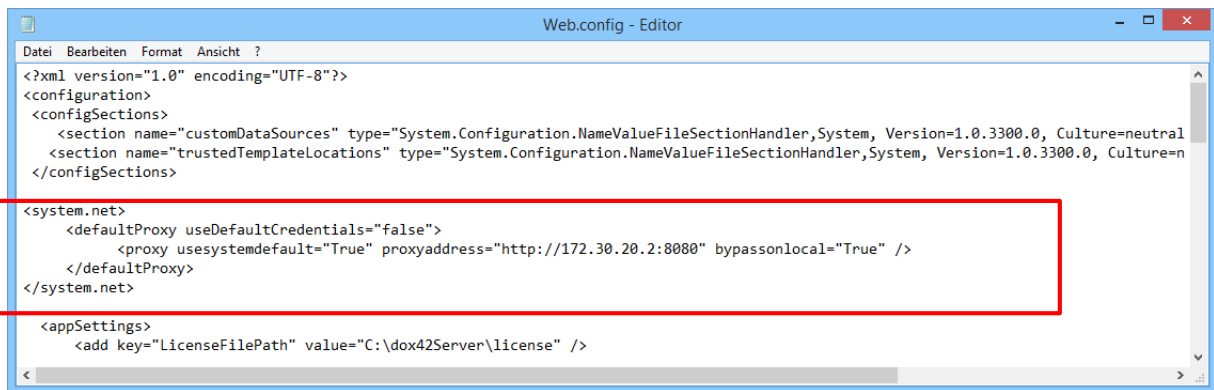
```
<system.web>
  <httpRuntime maxRequestLength="10000000" executionTimeout="1000"/>
</system.web>
```

License Activation

Please enter the license key in the Web.config as „LicenseKey“.

On the first call the dox42 Server will contact the dox42 license server to activate your key and write a license file (*.lic) in the folder indicated under “LicenseFilePath”.

If the Server is only allowed to access the internet via Proxy, maybe the following entry in the web.config will help.

A screenshot of a Windows Notepad window titled "Web.config - Editor". The window contains XML code for a Web.config file. A red rectangular box highlights the following configuration block:

```
<system.net>
  <defaultProxy useDefaultCredentials="false">
    <proxy usesystemdefault="True" proxyaddress="http://172.30.20.2:8080" bypassonlocal="True" />
  </defaultProxy>
</system.net>
```

The rest of the code in the window includes:

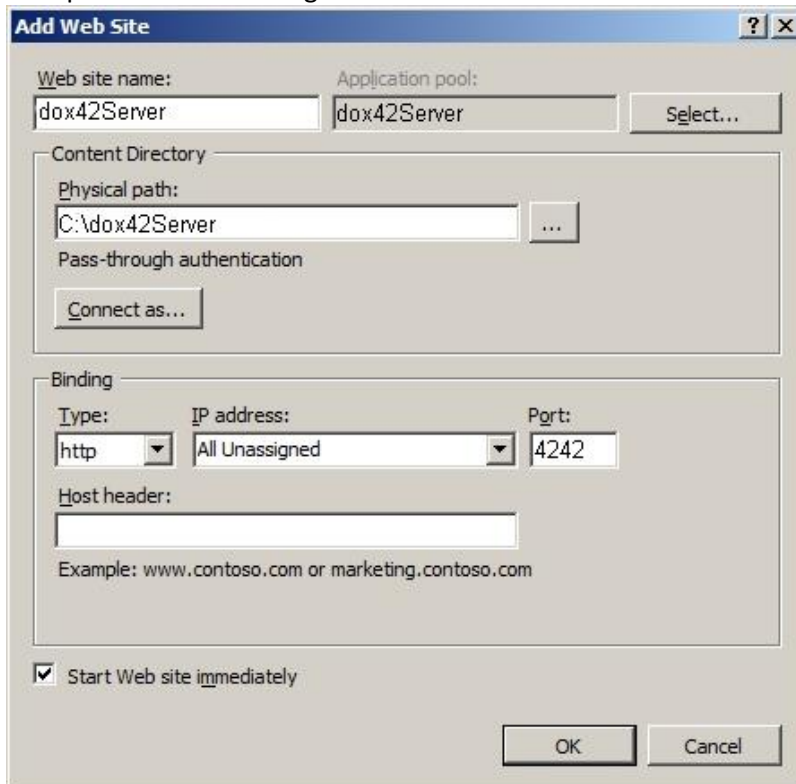
```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <configSections>
    <section name="customDataSources" type="System.Configuration.NameValueFileSectionHandler,System, Version=1.0.3300.0, Culture=neutral
  </configSections>
  <system.net>
    <defaultProxy useDefaultCredentials="false">
      <proxy usesystemdefault="True" proxyaddress="http://172.30.20.2:8080" bypassonlocal="True" />
    </defaultProxy>
  </system.net>
  <appSettings>
    <add key="LicenseFilePath" value="C:\dox42Server\license" />
  </appSettings>
</configuration>
```

In case you cannot establish an internet connection from your server, please contact support@dox42.com, we will provide you with your license file.

Installation

The following section describes the installation on a Windows 2008 R2 Server with IIS 7.5.

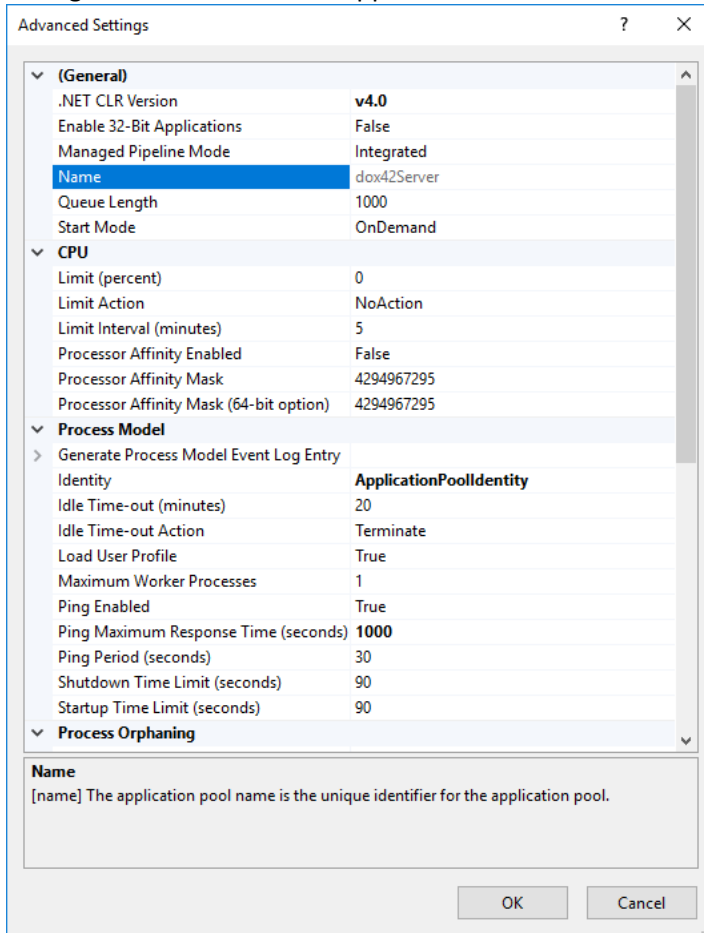
1. Copy the directory **dox42ServerPack\dox42ServerVXXXX\dox42Server** to **C:\dox42Server** (You can also choose a different virtual directory.)
2. Set up a site in the inetmgr:



Please mind the port! You may use any other port, but bear in mind that the standard port (80) is often already in use by other applications e.g. Sharepoint. Of course dox42 server supports https.

3. Copy your Aspose.Total.Lic file to the BIN Folder (C:\dox42Server\Bin) - not applicable for Trial Installations
4. Install the NET 4.5.2 framework.
5. User Server Manager to activate ASP.NET (inclusive HTTP Activation)
6. If want to enable access to SharePoint Online (Office 365) for this dox42 Server, please install the Microsoft SharePoint Server 2013 Client Components SDK.
Download: <http://www.microsoft.com/en-us/download/details.aspx?id=35585>
7. Run iisreset.

8. Configure the dox42 server-AppPool as follows:



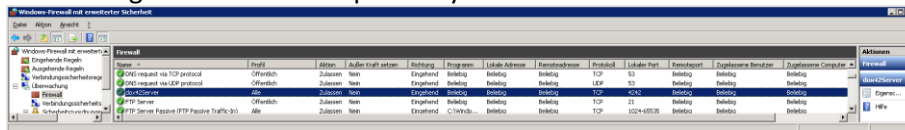
Instead of ApplicationPoolIdentity you can also use a Service-User.

In case you want to generate dox42 sample template which occasionally use MS Access (32-Bit needed) you might need to activate 32 bit applications.

9. In inetmgr, change back to site > manage permissions
 Here, look for the user "IIS AppPool\dox42Server" (Attention, use the local computer as search path!) or a dox42 service account and give this user read **and** write permissions to **C:\dox42Server**

10. Configure web.config (key, license-dir, mail server) and NLog.config (log-dir).

11. Do not forget to activate the port on your firewall.



12. Test with dox42 ServerTestClient, which can be found at C:\dox42ServerPackV32XX\dox42ServerTestClient_*

13. Be sure to use adequate Access Control to protect your dox42 Service against unauthorized use.

Upgrade

To Upgrade a dox42 Server please perform the following steps:

1. Make a backup of the dox42 Server folder, so you can roll back in case anything goes wrong.
2. Delete all files in the bin folder EXCEPT Aspose.Total.lic and all dll-files, that belong to your custom data sources and output actions. Custom data sources and output actions are registered in the web.config, e.g. dox42SAP, dox42AX, dox42DynamicCRM, etc. Please see the documentation of each module to get a list of the necessary libraries.
3. Copy all files from the bin folder of the dox42 Server Pack to the bin folder on your server.
4. Copy all files EXCEPT dox42LIVE.config from the dox42LIVE folder of the dox42 Server Pack to the dox42LIVE folder on your server.
5. Check if Dox42Service.asmx and dox42RestService.ashx are present in your dox42 Server folder.
6. Check your web.config for new configuration options.
7. Restart your dox42 Server in IIS.
8. Test with dox42 ServerTestClient
9. That's all, you are done, go for a beer 😊

Support

Should you have any questions, please do not hesitate to contact support@dox42.com

Good luck with your dox42 server!