
Setup Guide Automation AIS-Create Document Service v2.1

Updated: Date 2022-06-30

Contents

1	Installation Guide	3
1.1	Pre-installation instructions.....	3
1.2	Environmental Configuration	4
1.3	Configuration Settings	4
1.4	Additional Settings.....	7
1.5	License Key	8
1.6	Simple distribution – ‘Get batch’	10
1.7	Automatic installation of scripts – PL*SQL	12
1.8	Customized setup	14
1.9	Disk Space Requirements	16
1.10	Ready to Install	18
1.11	Configure the web service on IIS	19
1.12	IFS Applications - IFS integration configuration.....	20
1.13	IFS Cloud - IFS integration configuration	21
1.14	Trigger Service request using WcFTestClient	23
1.15	IFS Applications – Mapping data to templates	27
1.16	IFS Cloud – Mapping data to templates	27
1.16.1	Mapping file explained	27
1.16.2	Mapping singular document properties – standard.....	28
1.16.3	Mapping singular document properties – using quick reports	33
1.16.4	Mapping tables.....	35
2	Upgrade notes.....	40
2.1	Upgrading old versions to Automation Server 2.1. 33715 or higher from older versions.	40
3	Update AIS CreateDocumentService via zip file.....	40

1 Installation Guide

1.1 Pre-installation instructions

Before the module is uninstalled, changed, or repaired, make sure that config files in the Directory and web config file with related message Tracking folder in inetpub are backed up properly.

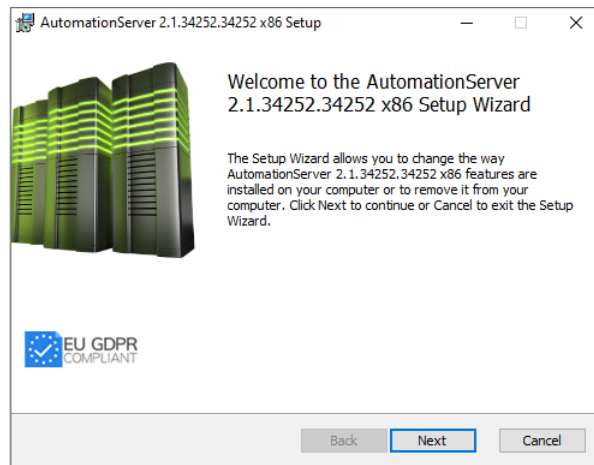


Figure 1 – Setup Wizard

The Automation Server with all the modules is normally installed through an MSI package. Run the .msi file and follow the installation wizard. It can also be directly deployed as a virtual application in IIS using just a zip file, but then manual steps are required such as adding the license in the registry hive. The following steps are when using MSI installation:

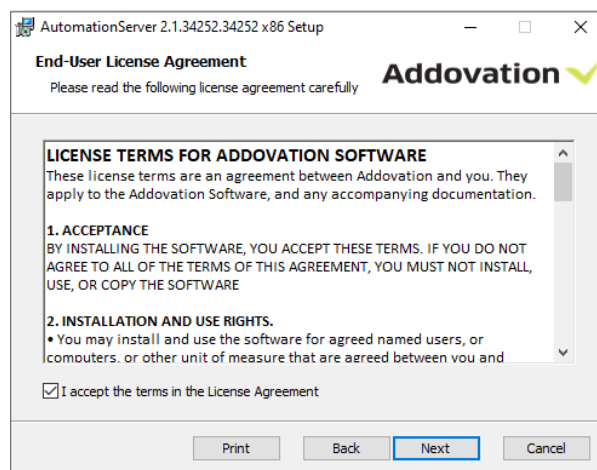


Figure 2 – End-User License Agreement

1.2 Environmental Configuration

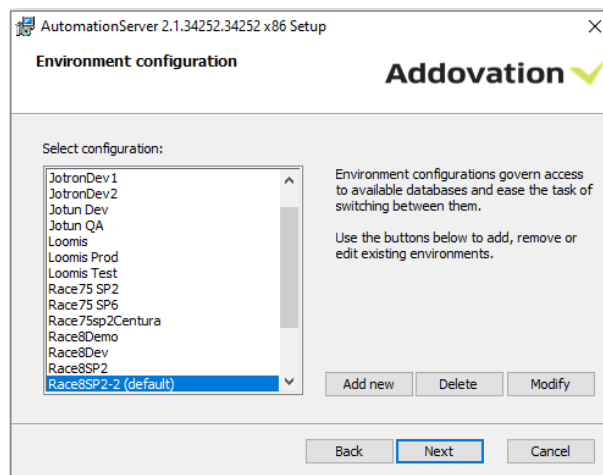


Figure 3 – Environmental Configuration

When installing the software for the first time, you will need to specify which environment(s) you wish to be able to connect to. You do this by clicking the **Add new** button. When upgrading to a new version, or modifying an existing installation, you may choose to retain the current settings and simply click Next or make changes to the environment configuration by clicking Modify.

You will be able to choose between the environments in the AIS login dialog, from any of the supported MS Office applications.

1.3 Configuration Settings

The configuration settings allow you to customize your installation.

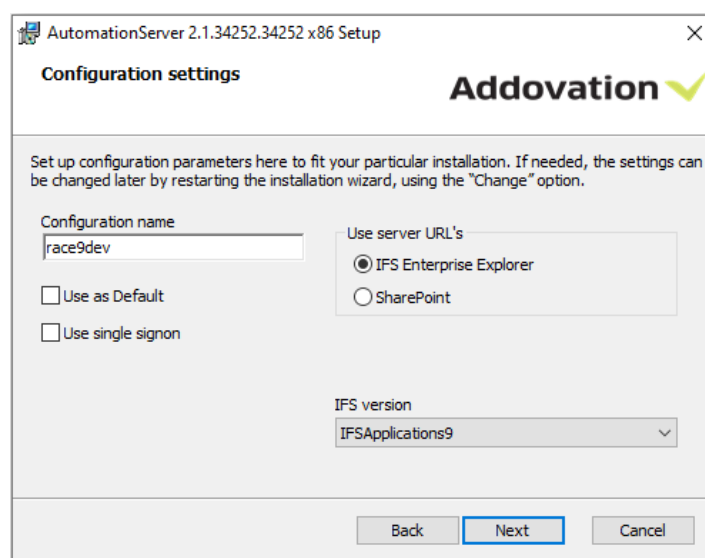


Figure 4 - Configuration Settings

Configuration name: Choose a name that describes the configuration (e.g. Race 8 demo), if it should be the default logon choice, and whether IFS is configured for single sign-on.

Server URLs: Select the server URL of your choice

IFS Version: Choose your IFS version

If you are not sure what options to check, leave the default values. You can change the values temporarily from within the application at any time.

To add a new environment, you need to specify the address to IFS Extended Server (e.g. server1:58080), the Application owner (e.g. IFSAPP) and the IFS portal (e.g. <http://race75sp6:58080/>).

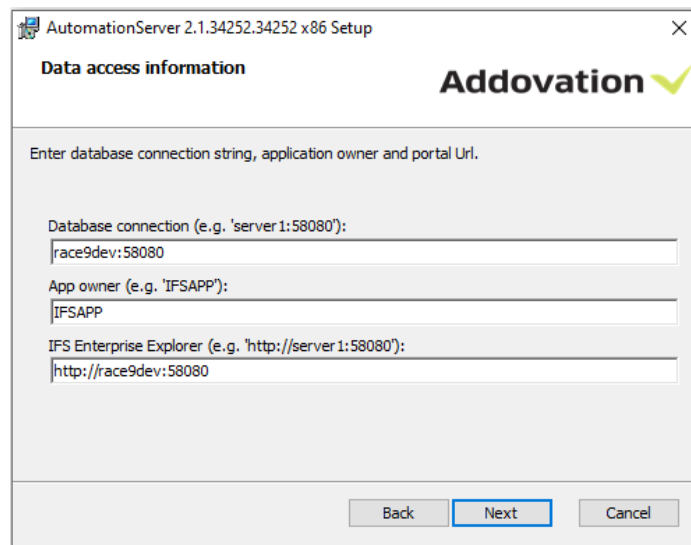


Figure 5 - Data Access Information

The Database connection must use the prefix HTTP or HTTPS (in lower-case).

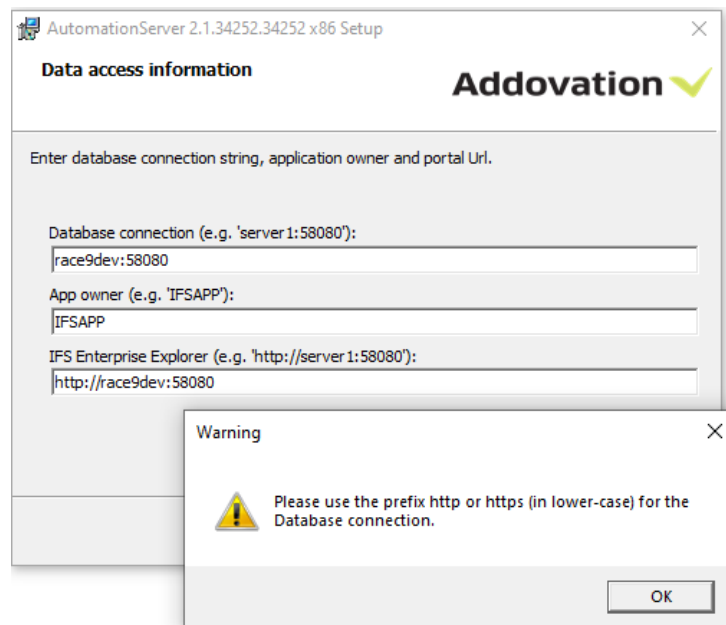


Figure 6 - Data Access Information

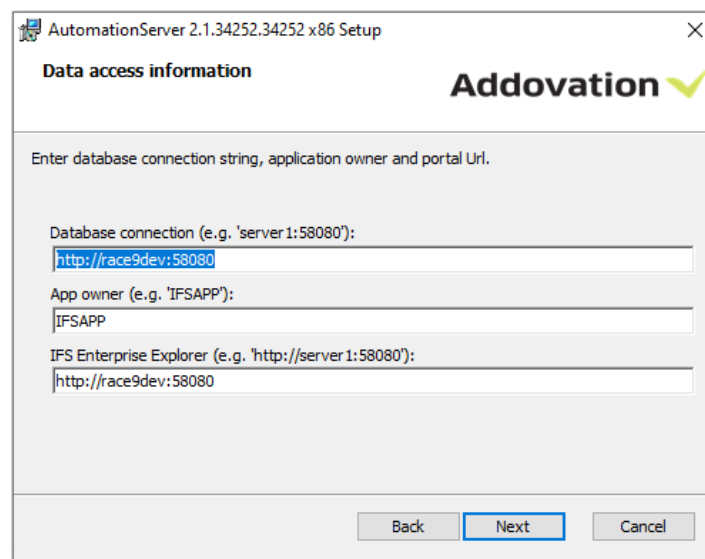


Figure 7 - Data Access Information

1.4 Additional Settings

In the Additional Settings dialog, you can define a symbol that exclude classes and formats with this particular symbol. This field is defaulted to the asterisk symbol. You can also define the Document Search URL in this dialog.

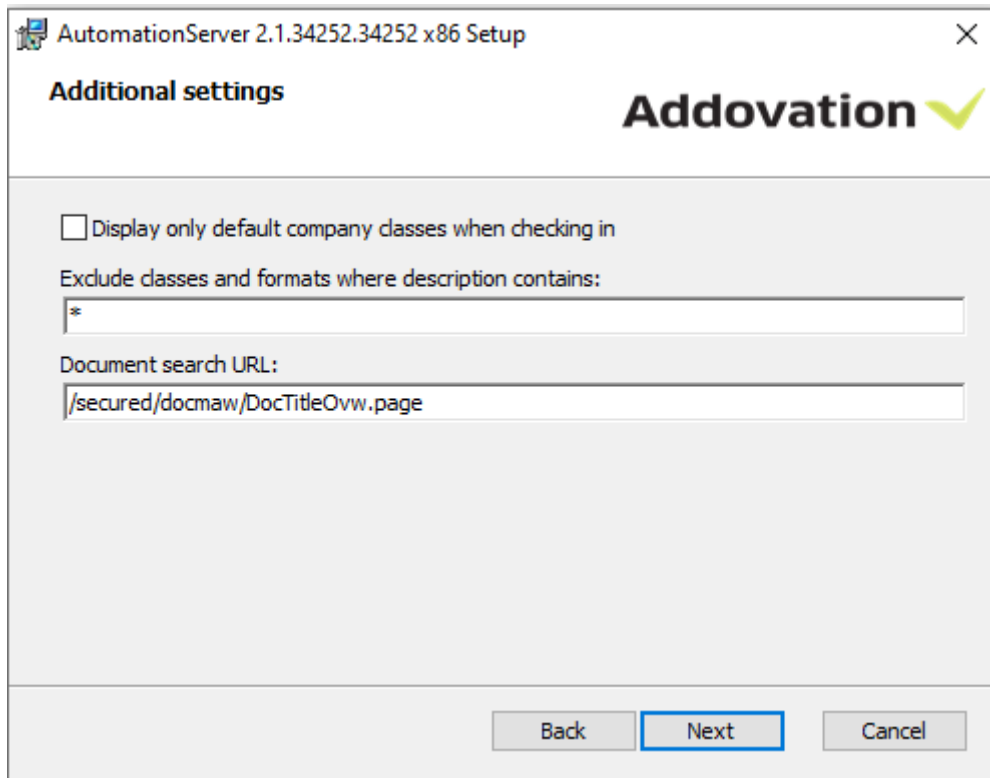
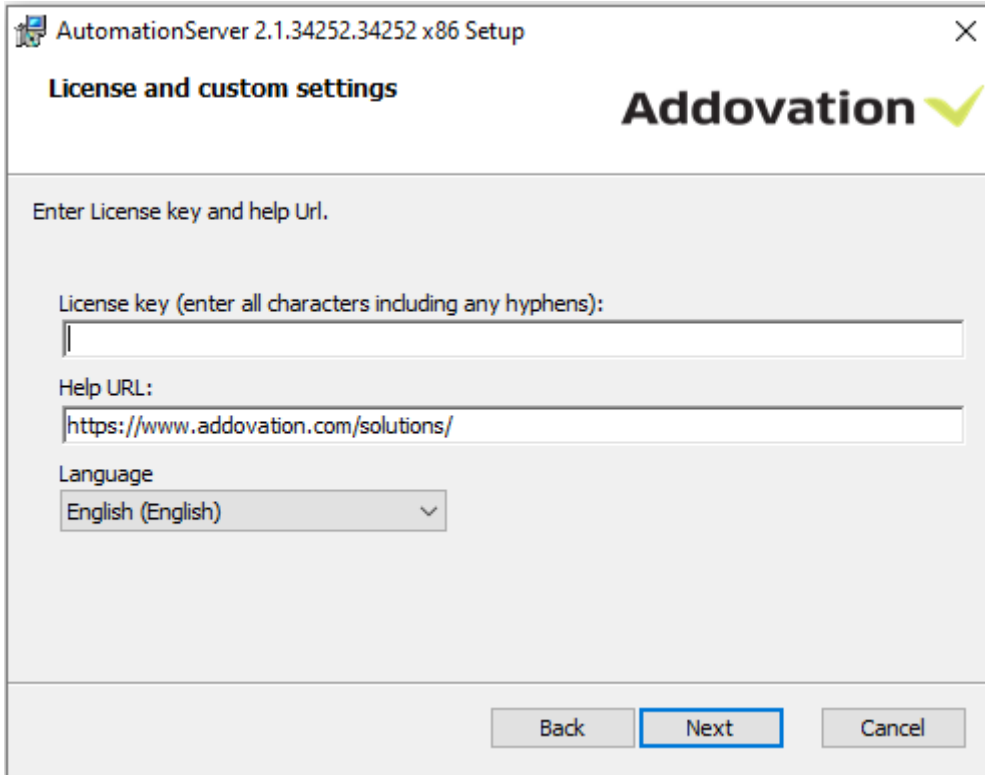


Figure 8 – Additional Settings

1.5 License Key

You will have received a license key from Addovation upon purchase of the AIS. Please enter the license key during the configuration of the software.



AutomationServer 2.1.34252.34252 x86 Setup

License and custom settings

Addovation ✓

Enter License key and help Url.

License key (enter all characters including any hyphens):

Help URL:

Language

English (English)

Back Next Cancel

Figure 9 - License and custom settings

In the Help URL field, you can specify what the help symbol in the AIS will point to. Included in the AIS is an online user documentation. The link is normally already filled in, so if you don't want to provide your own information, you can leave it as is.

When you have entered the correct license key, you will receive a validation listing the products and their date of expiry.

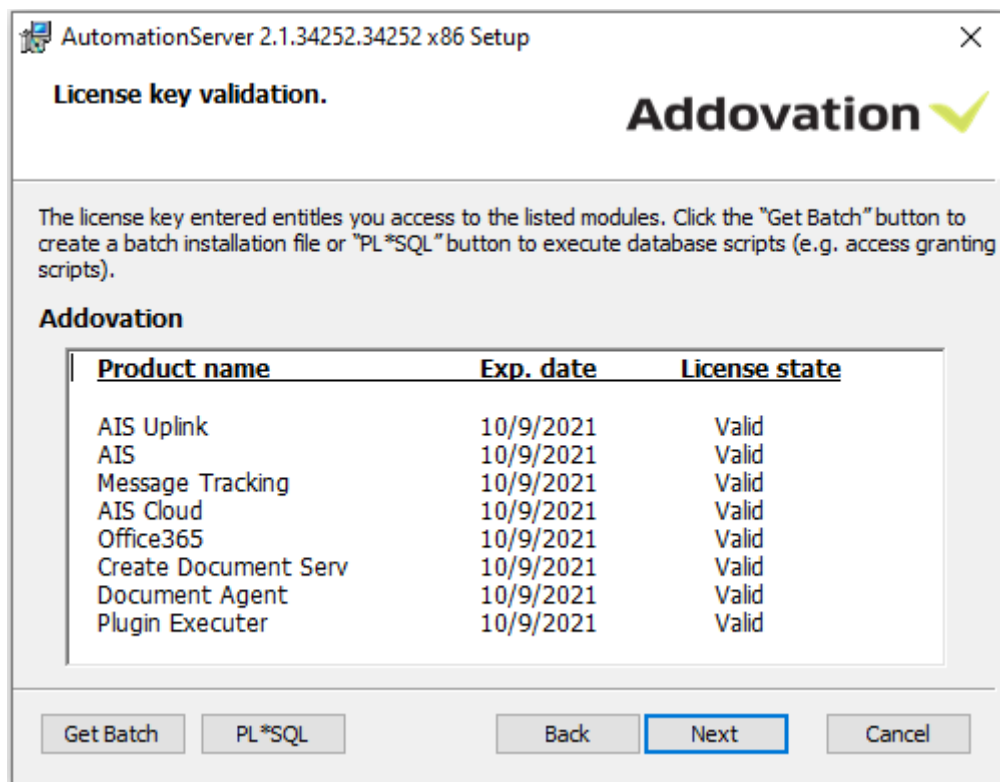


Figure 10 – License key validation

1.6 Simple distribution – ‘Get batch’

The ‘Get batch’ option allows you to install the software on several machines, without having to enter your settings multiple times.

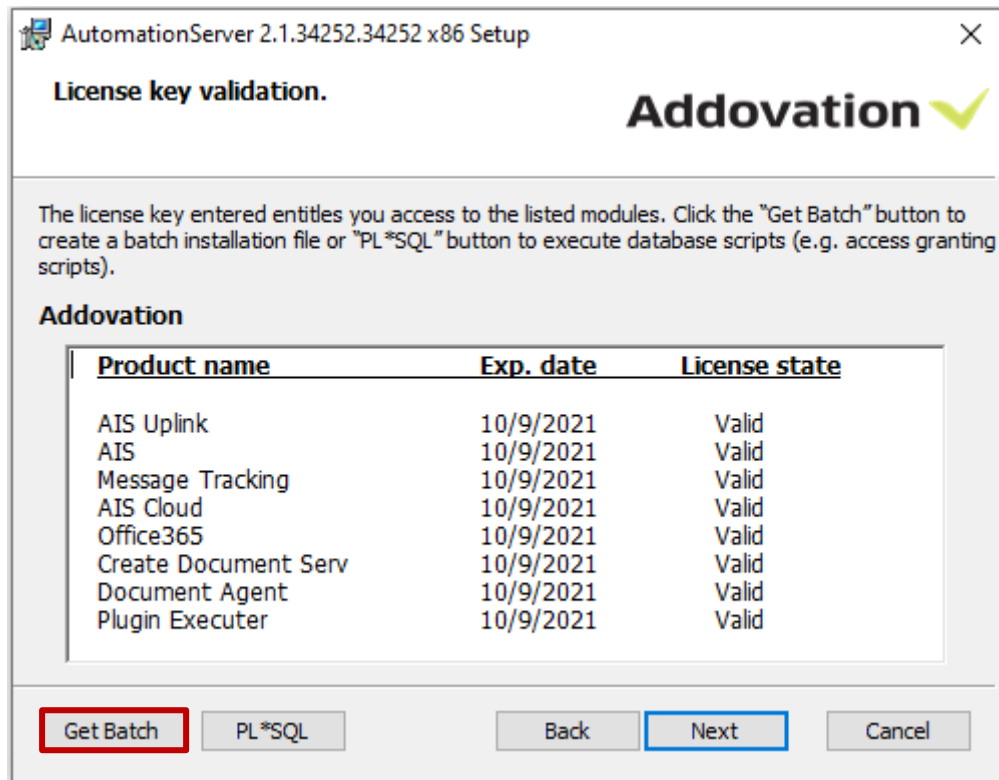


Figure 11 – Get Batch

When you click on the button ‘Get Batch’, a .bat file is created, which will launch the .msi using the settings you defined in the current installation.

Observe the following with generating the batch file:

- The batch file must be executed by a user having administrator rights
- If you experience problems running from a network drive, try to copy the installation file to a local drive
- The batch file created can only be used to install 32 bit or 64-bit version, depending on the installation file.

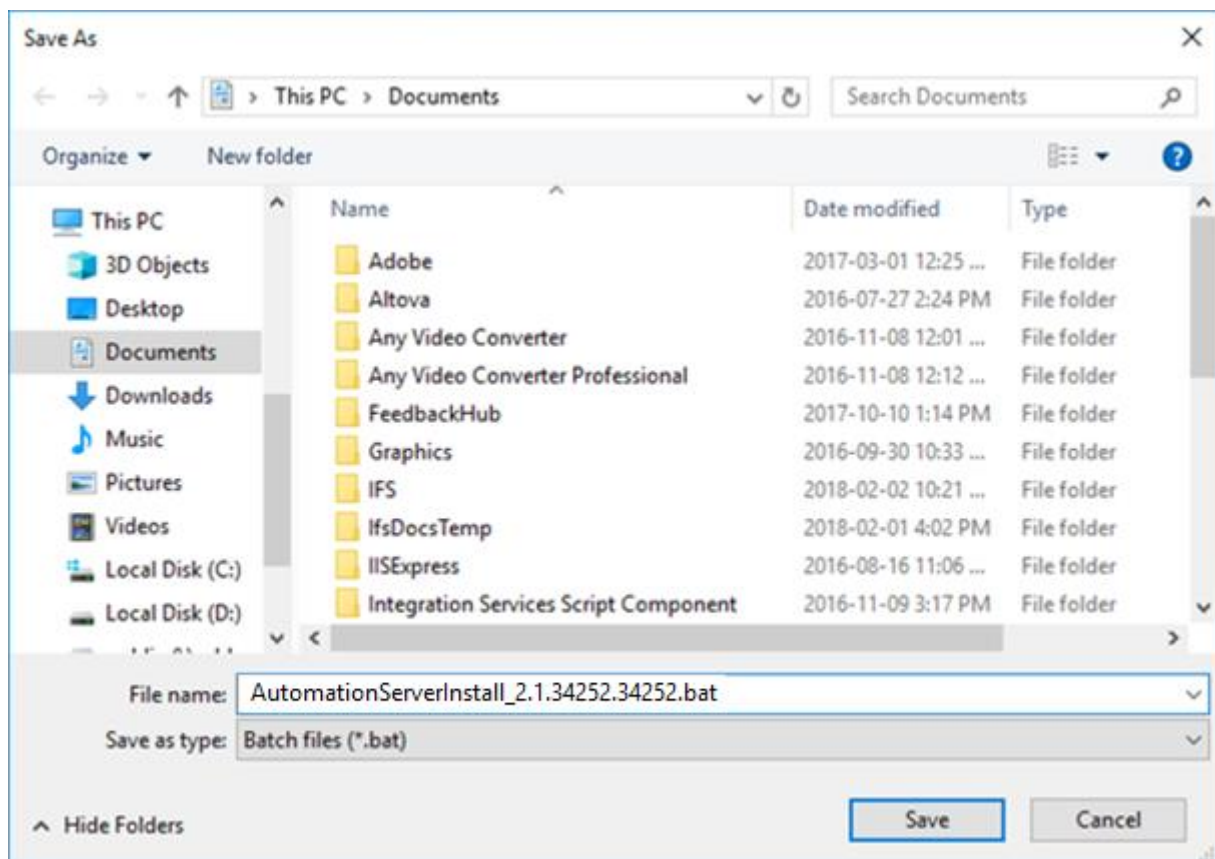


Figure 12 – Save batch file

A .bat file and a .PS1 file is generated at your chosen location.

When you want to distribute the software to other computers, download the .msi, the.bat and the .PS1 file to the chosen computer. Open the .bat file and right click 'run as administrator'. The installation file will then run with your chosen parameters, installing AIS on the computer.

1.7 Automatic installation of scripts – PL*SQL

If this is the first time you run the AIS installation, you need to run the PL*SQL function.

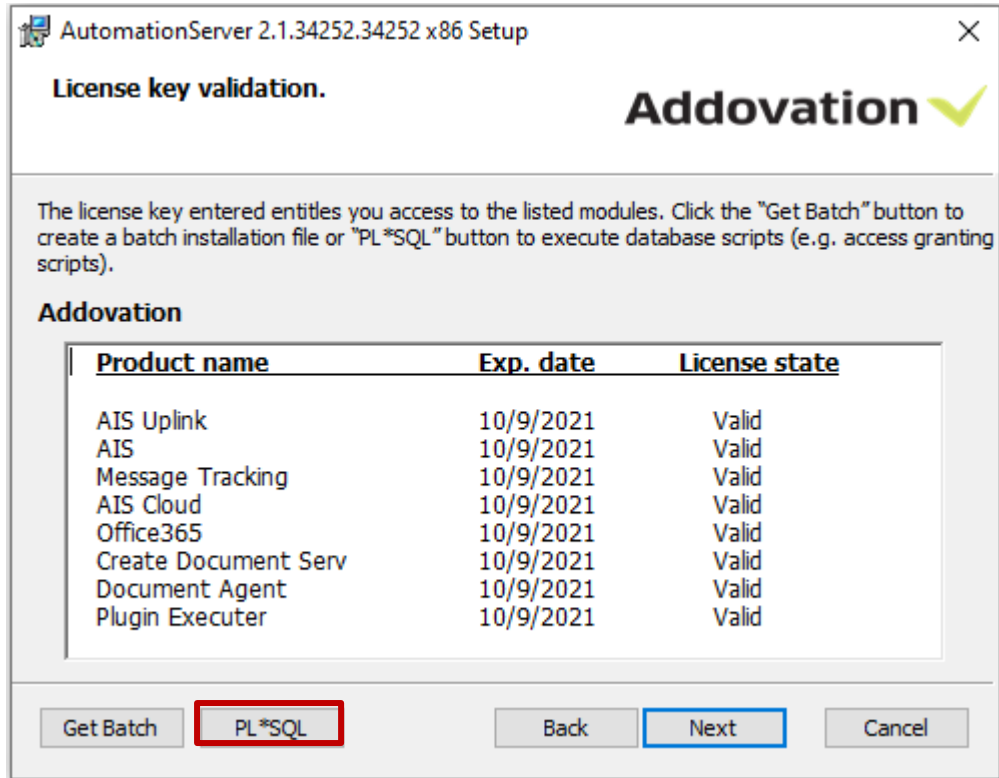


Figure 13– PL*SQL script

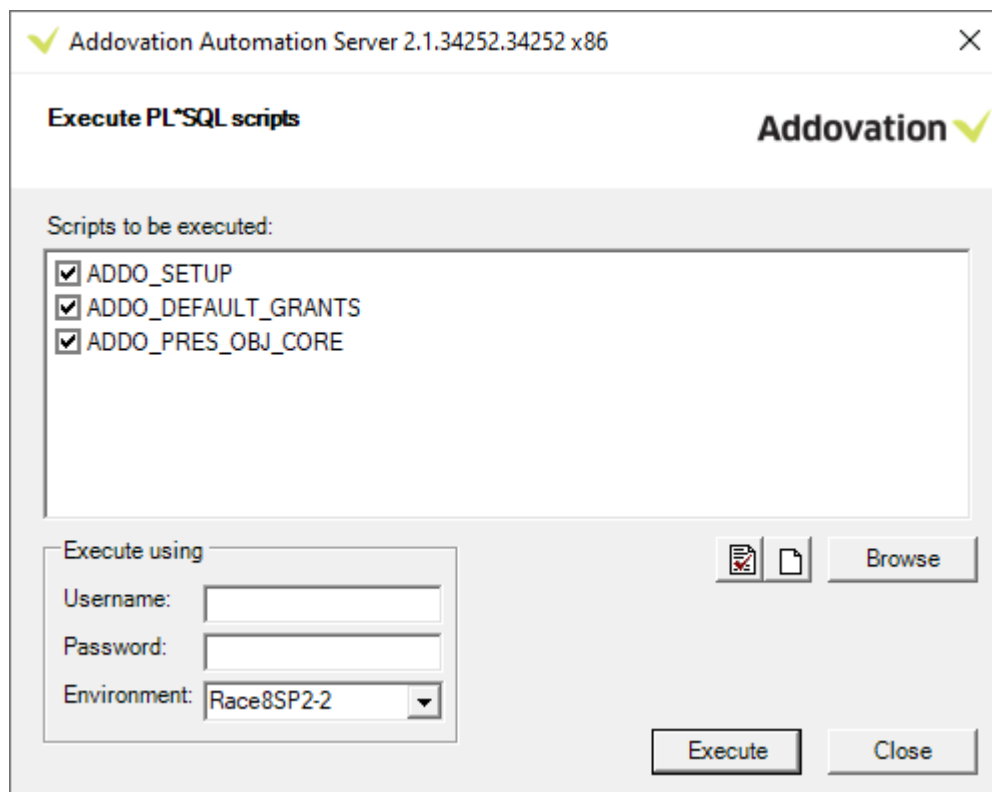


Figure 14 – Execute PL*SQL scripts

Click on the PL*SQL button and type in your username and password (Usually IFSAPP or IFSADM) and the environment where you would like to install the scripts.

If you have scripts saved to your drive, you can browse for these using the 'browse' button. Click 'Execute' to run the scripts and to give the necessary access and grants. The wizard will confirm that the scripts were run successfully. You may have to enter parameters for some of the scripts

1.8 Customized setup

Under 'custom setup' you can select the way the features are to be installed for the various MS Office components. Click on the icons in the tree list to adjust the installation for the components.

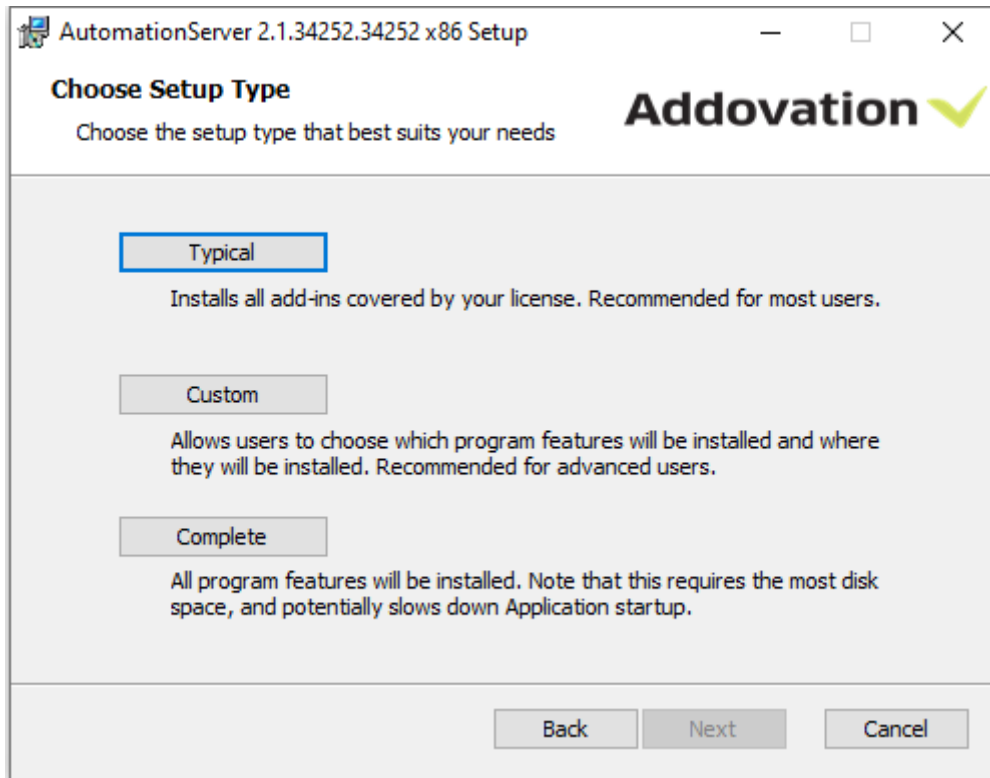


Figure 15 – Setup Types

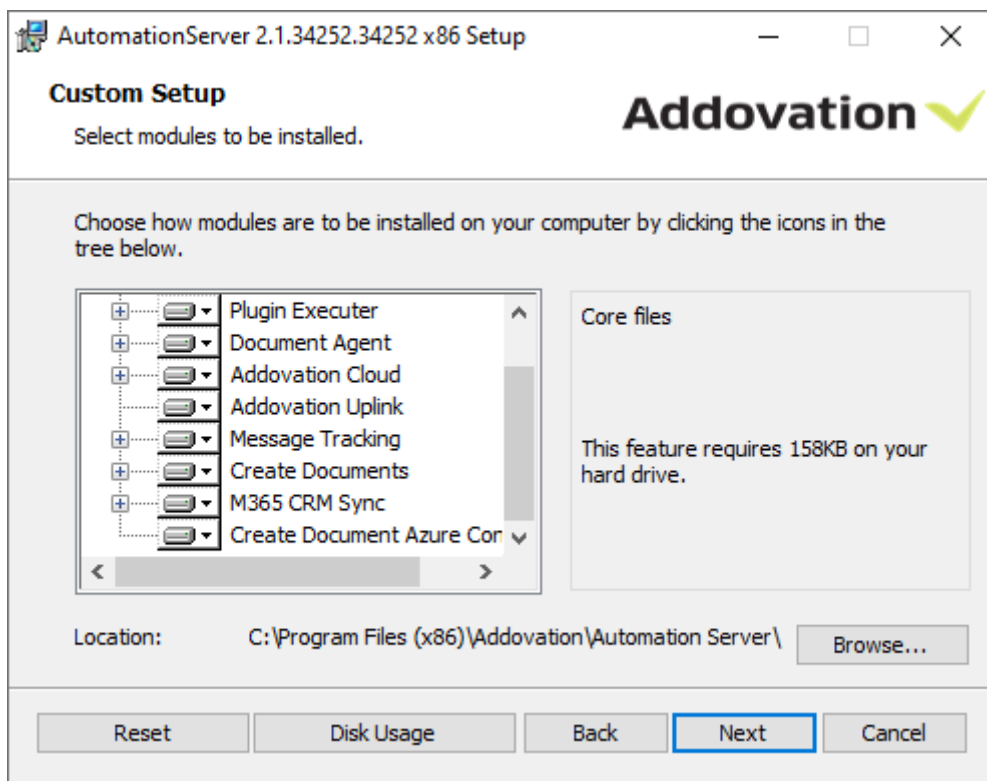


Figure 16 – Custom Setup

Components

- IFS to SQL Server Plugin
- Plugin Executor
- Document Agent
Reads files or emails from configured folder or mailbox and does stuff like create work orders or other IFS items.
- Addovation Cloud
- Addovation Uplink
- Message Tracking
- Create Documents
- M365 CRM Sync
- Create Document Azure Consumer

Location

Always try to use default location if possible. It's easier to find the folders if they are where they should be.

1.9 Disk Space Requirements

According to the selected features in the above step, disk space requirement is calculated and displayed in the Required Column. You can see this by pressing the Disk Usage.

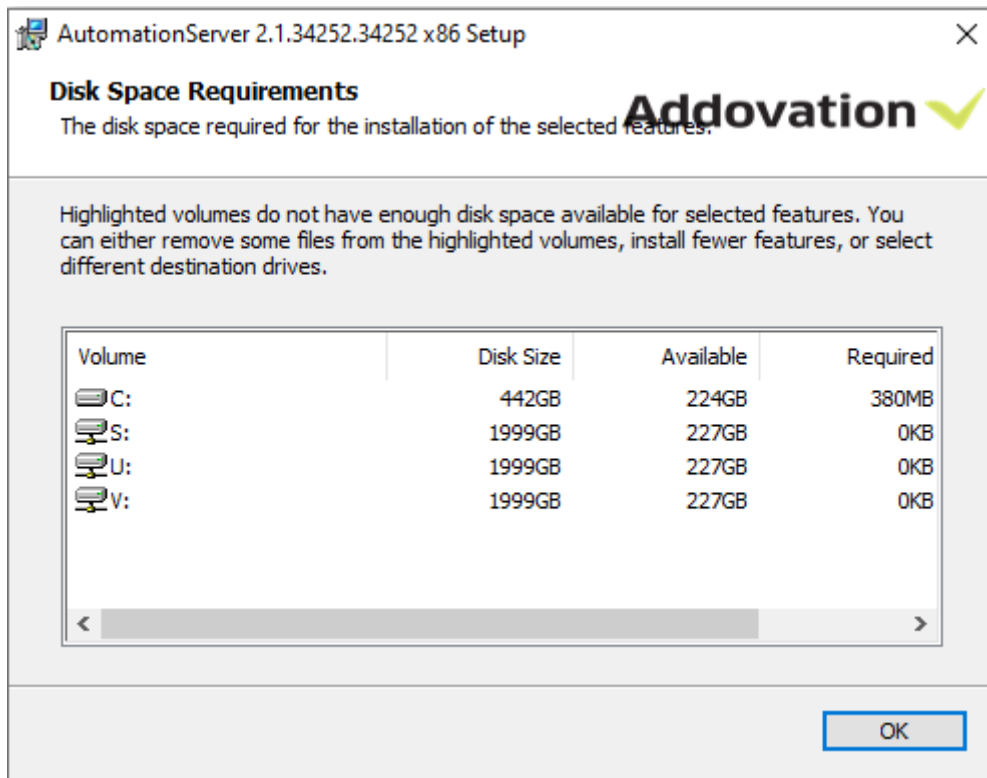


Figure 17 – Disk Space Requirements

Create Documents Service is not be able to install/upgrade without IIS. Setup will exit after giving below warning message.

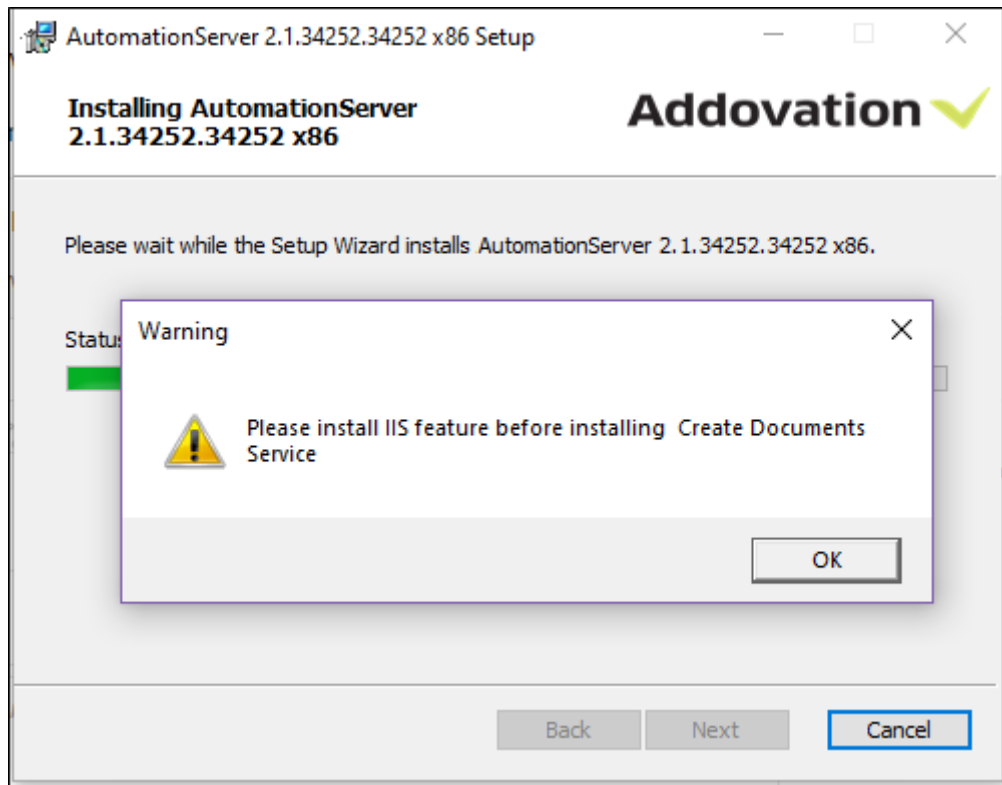


Figure 18 - IIS requirement

1.10 Ready to Install

After adjusting the disk space as per the need in your local machine you can complete the installation

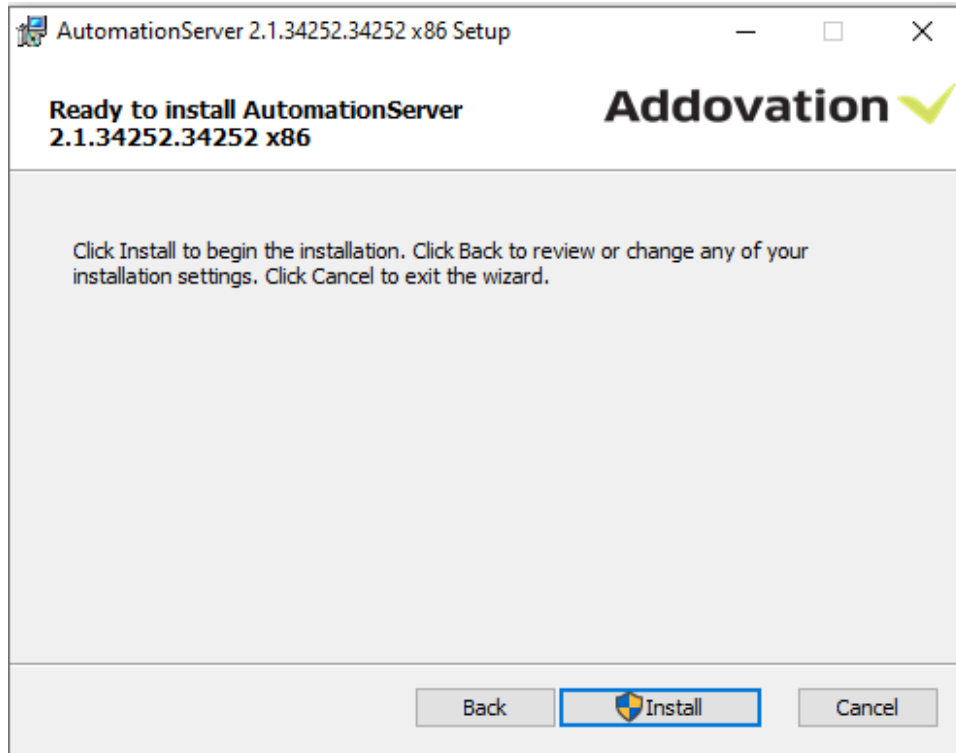


Figure 19 - Install

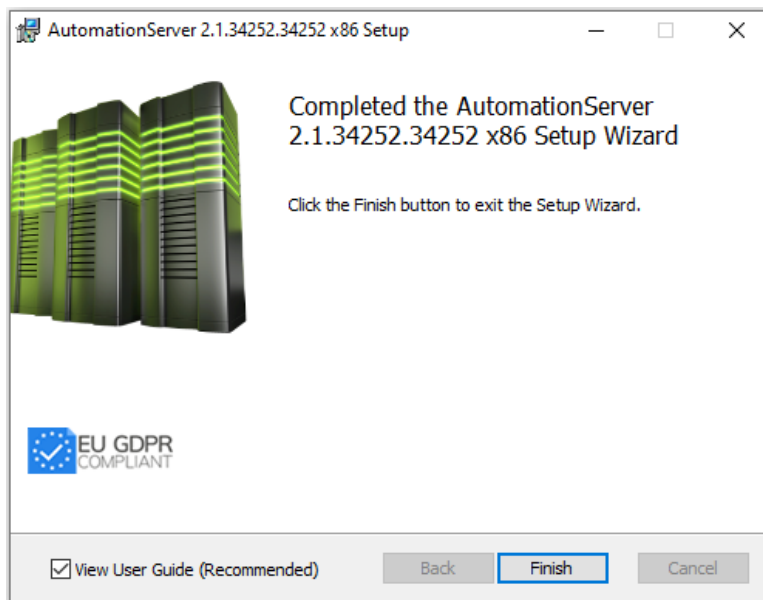


Figure 20 - Finish

1.11 Configure the web service on IIS

- Once the create document module is installed, 'CreateDocuments' folder will be created in the 'inetpub' folder.

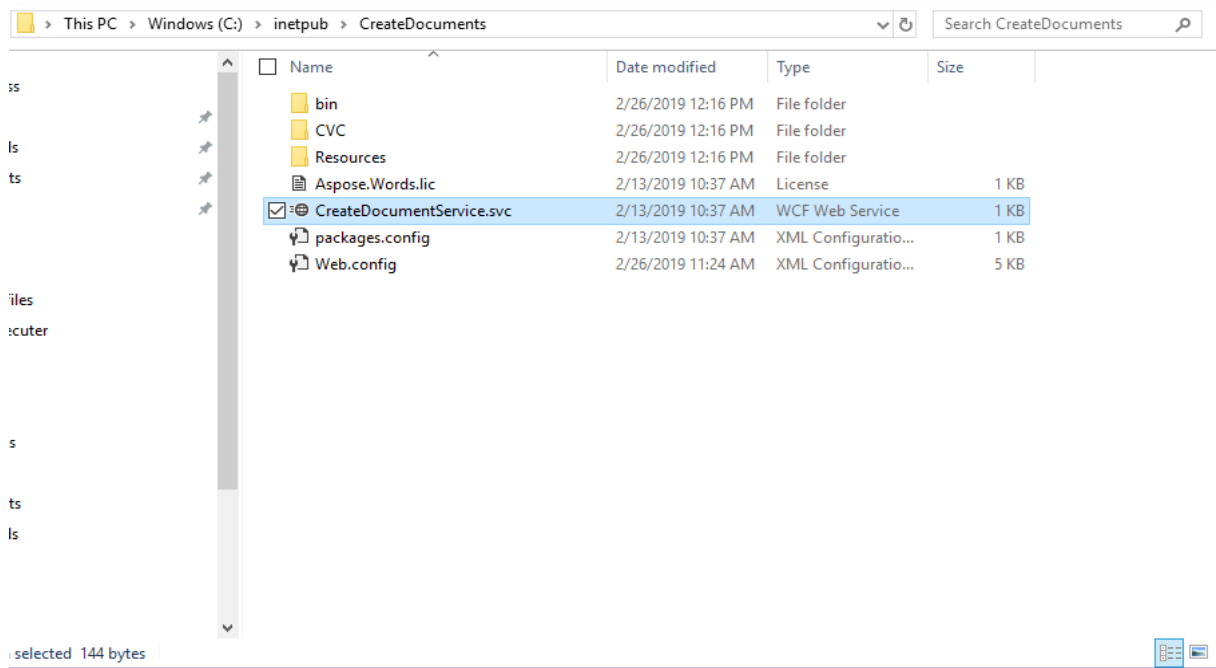


Figure 21 - Create Document in 'inetpub' Folder

- IIS configuration to host WCF Service

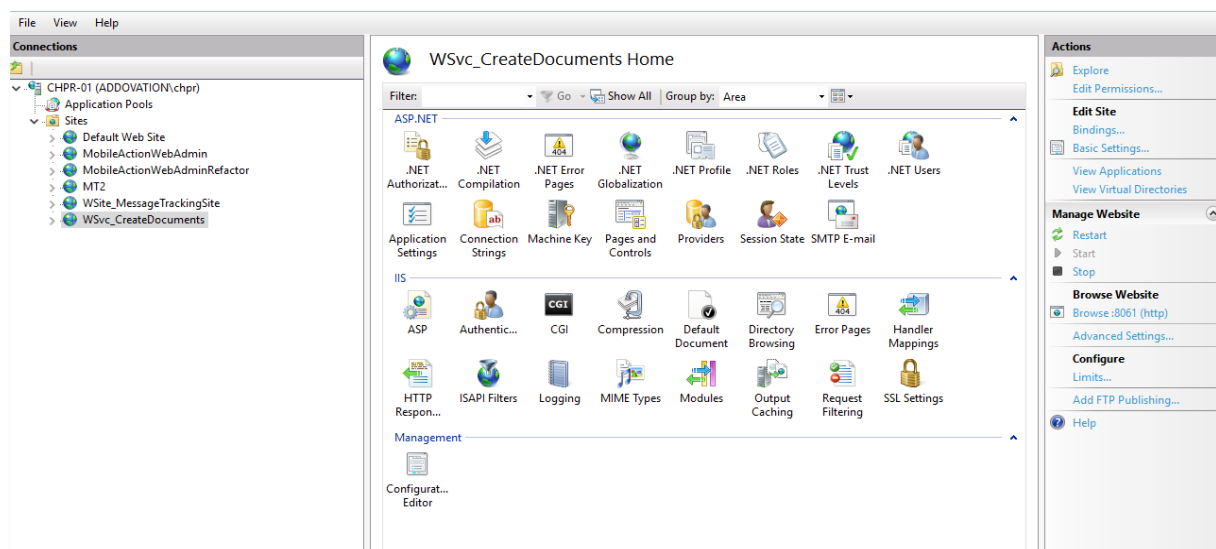


Figure 22 - IIS with installed site

- If the service is installed successfully service can be browsed as depicted below.

CreateDocumentService Service

You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the `svcutil.exe` tool from the command line with the following syntax:

```
svcutil.exe http://localhost:8061/CreateDocumentService.svc?wsdl
```

You can also access the service description as a single file:

<http://localhost:8061/CreateDocumentService.svc?singleWsdl>

This will generate a configuration file and a code file that contains the client class. Add the two files to your client application and use the generated client class to call the Service. For example:

C#

```
class Test
{
    static void Main()
    {
        CreateDocumentServicePortTypeClient client = new CreateDocumentServicePortTypeClient();

        // Use the 'client' variable to call operations on the service.

        // Always close the client.
        client.Close();
    }
}
```

Visual Basic

```
Class Test
  Shared Sub Main()
    Dim client As CreateDocumentServicePortTypeClient = New CreateDocumentServicePortTypeClient()
    ' Use the 'client' variable to call operations on the service.

    ' Always close the client.
    client.Close()
  End Sub
End Class
```

Figure 23 - Browse service

1.12 IFS Applications - IFS integration configuration

- Web config file should be modified by providing values to point the service to the correct IFS Applications instance as depicted below.

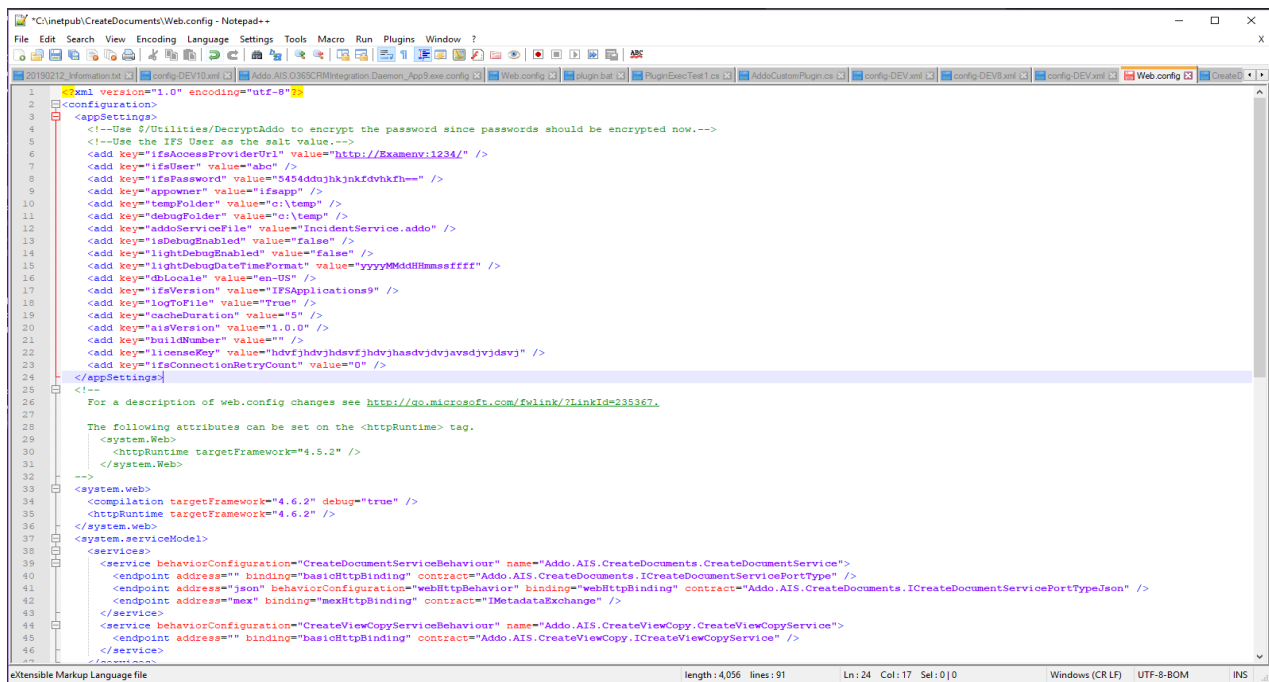


Figure 24 - Create Document Web Config file

"ifsAccessProviderUrl"	Point to the application server hosting IFS Applications.
"ifsUser"	The integration user used in accessing IFS Applications. The database access authorization rights generated under the subheading "Generate database access authorization script" should be granted to this user.
"ifsPassword"	The password for the "ifsUser".
"appowner"	The name of the application owner for the IFS Application instance
"tempFolder"	A temporary folder to store temporary files. This location should be cleaned regularly.
"debugFolder"	Folder where debug information will be stored
"isDebugEnabled"	has the options Off, Light and Full Off – No debug information will be created Light – Details related to performance will be generated Full – Details related to performance as well as other information on the tasks performed will be stored in order to assist in debugging
"dbLocale"	DB local of the database.
"numberConnections"	Number of concurrent connections created to access the database for certain activities.
"ifsVersion"	The major version of IFS Applications.
"templateCacheDuration"	The duration that the templates will be cached in memory.

1.13 IFS Cloud - IFS integration configuration

For IFS Cloud, some new web.config parameters has been introduced. In web.config, in order to configure URL and integration user, the following keys need to be set:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <configuration>
3   <appSettings>
4     <!--Connection details for IFS Applications-->
5     <!--IFSAccessProviderUrl defines the access point for the IFS applications instance that the service should connect to-->
6     <add key="ifsAccessProviderUrl" value="https://pagcloudtest20220613.pagcloud.com" />
7     <!-- e.g. /auth/realms/<REALM_HERE>/protocol/openid-connect/token -->
8     <add key="ifsTokenEndPoint" value="/auth/realms/pagcloudtest/protocol/openid-connect/token" />
9     <!-- Folder in where mapping files between template and data resides -->
10    <add key="mappingFolder" value="C:/Addovation/20220613" />
11    <!-- As registered in IFS Cloud21R2, disregarded if IFS Applications 10 -->
12    <add key="ifsClientId" value="TSM_ADDOVATION20220613" />
13    <!-- As registered in IFS Cloud21R2 and encrypted, disregarded if IFS Applications 10 -->
14    <add key="ifsClientSecret" value="C:\Users\TSM\Documents\Addovation\20220613\Addovation20220613.ifsClientSecret" />
15    <add key="ifsUser" value="" />
16    <!--Use $/Utilities/DecryptAddo to encrypt the password since passwords
17    should be encrypted now. Use the IFS User as the salt value.-->
18    <add key="ifsPassword" value="" />
19    <!-- Typical value for Appowner is IFSAPP. This may change depending on
20    the installation of IFS Applications.-->
21    <add key="appowner" value="" />
22    <!--tempFolder is used as a repository to hold all temporary files used
23    by the service.-->
24    <add key="tempFolder" value="C:/Addovation" />
25    <!--Debug settings-->
26    <!--debugFolder defines the folder where debug log files will be stored.-->
27    <add key="debugFolder" value="C:/Addovation" />

```

ifsAccessProviderUrl	The URL to the system, normally it is https://someurl.com.
ifsTokenEndPoint	The token end-point that has been configured for the IFS Cloud instance.
mappingFolder	The folder in where all mapping files are being kept and maintained.
ifsClientId	As defined in IFS Cloud IAM.
ifsClientSecret	As defined in IFS Cloud IAM and encrypted using Addovation Encryption.

The following keys can be omitted when going against an IFS Cloud instance: ifsUser, ifsPassword, appowner.

Last, we need to tell the service that this is an IFS Cloud instance:

```

C:\inetpub\wwwroot\ais\pagcloudtest\20220613\Web.config - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
34 <!--lightDebugDateTimeFormat uses a C# date time format to format
35 <add key="lightDebugDateTimeFormat" value="yyyyMMddHHmmssffff" />
36 <add key="logToFile" value="True" />
37 <add key="dbLocale" value="en-US" />
38 <!--ifsVersion defines the version of IFS applications that the ser
39 <!--add key="ifsVersion" value="IFSApplications9"/-->
40 <add key="ifsVersion" value="IFSCloud21R2" />
41 <add key="cacheDuration" value="5" />
42 <add key="ifsConnectionRetryCount" value="0" />
43 <add key="aisVersion" value="1.0.0" />
44 <add key="buildNumber" value="" />
45 <add key="licenseKey" value="" />
46 <add key="addoServiceFile" value="IncidentService.addo" />
47 </appSettings>
48

```

Note: Even though the IFS Cloud instance is running IFS Cloud 22R1, "IFSCloud21R2" can and must be used. IFS Cloud 21R2 and IFS Cloud 22R1 is using the same authentication flows.

1.14 Trigger Service request using WcFTestClient

- Run WcFTestClient resides in visual studio root folder.

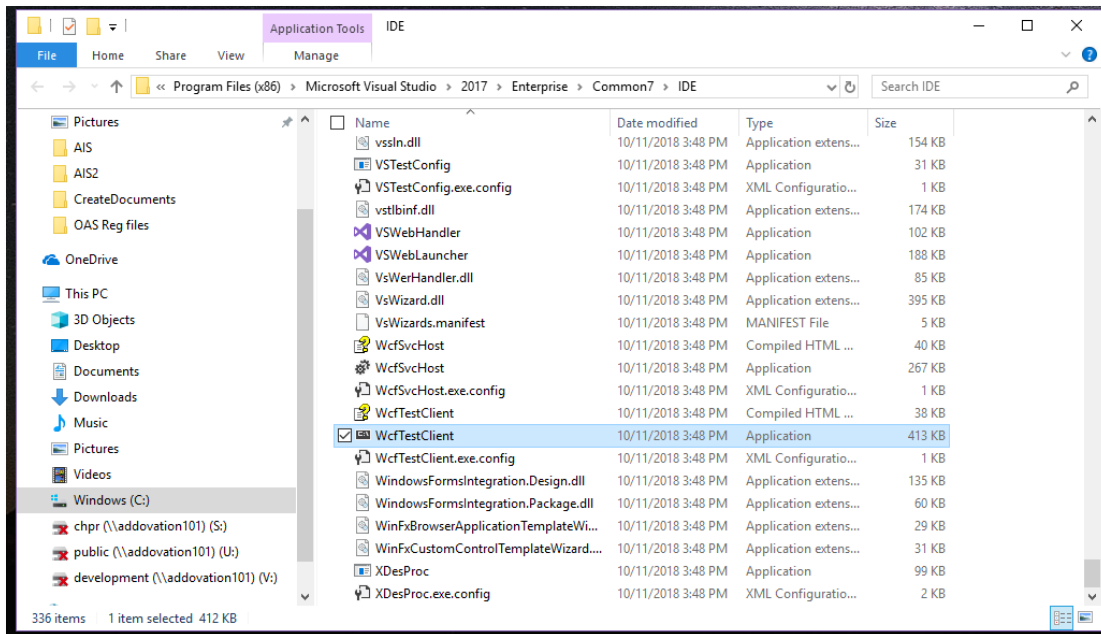


Figure 25 - WcFTestClient

- Provide the address of endpoint service

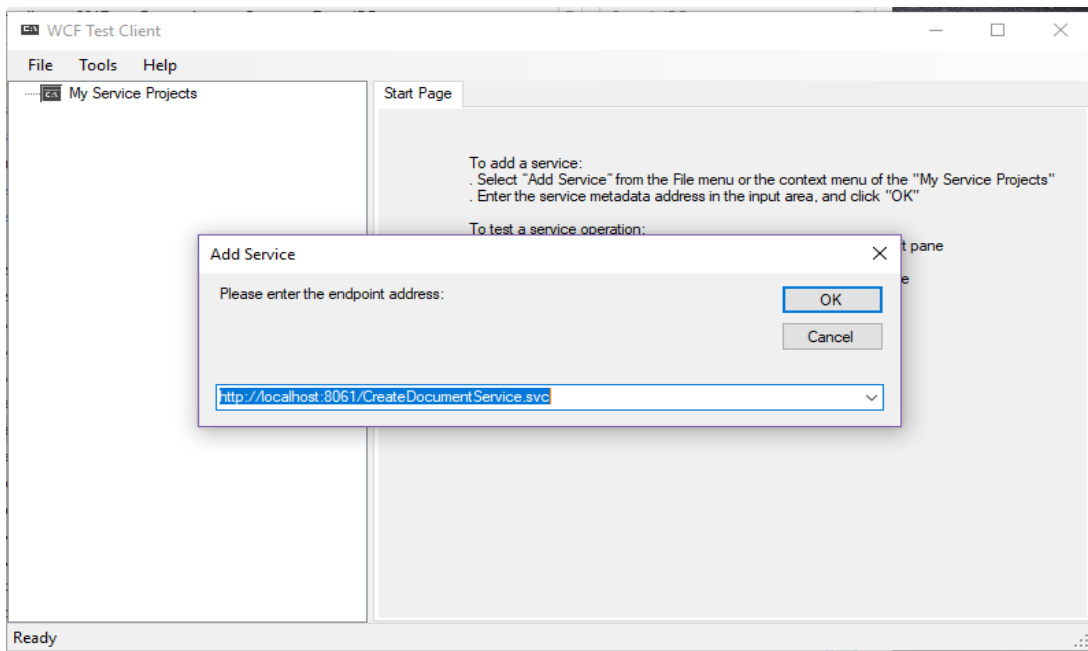


Figure 26 - Add service address

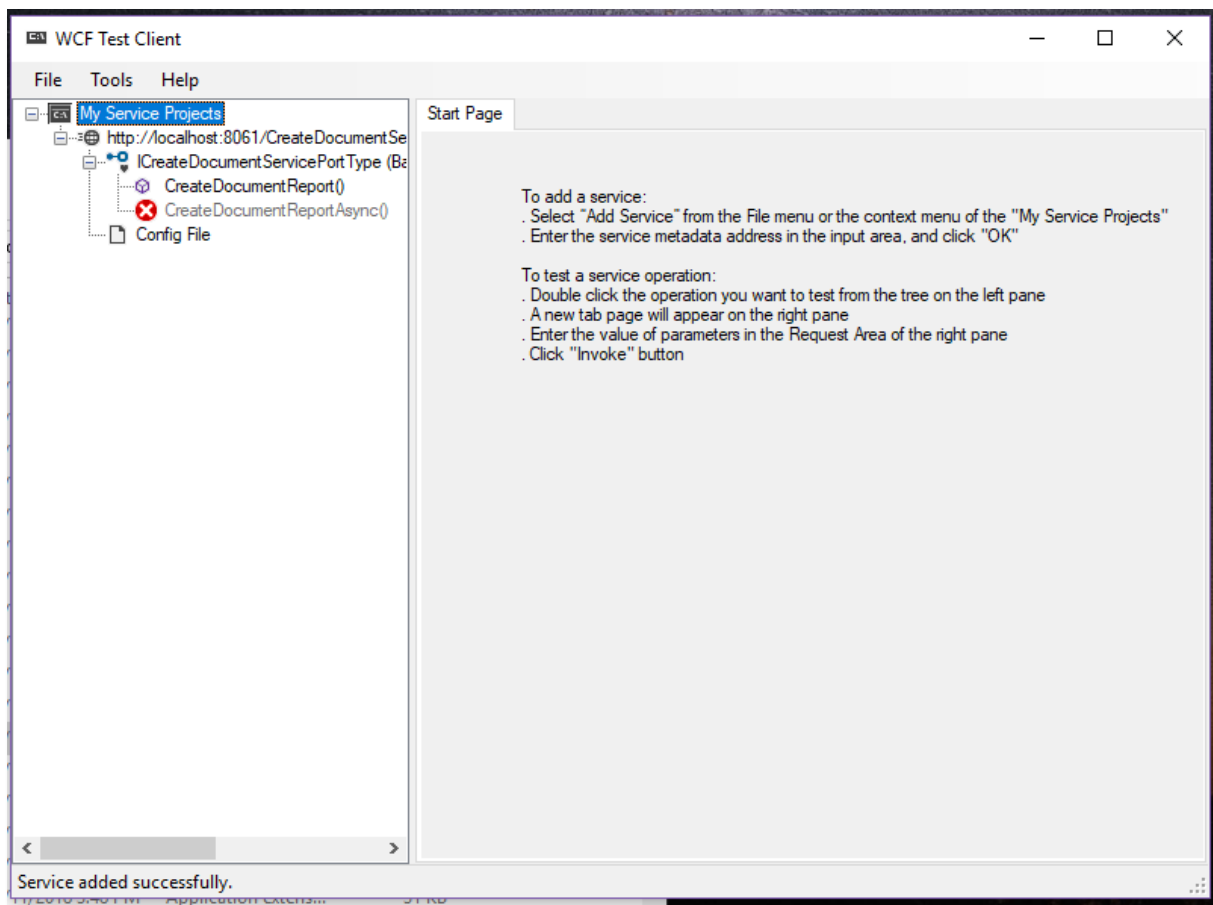


Figure 27 - Service Projects in WcFTestClient

- Double click on CreateDocumentReport() related to given service

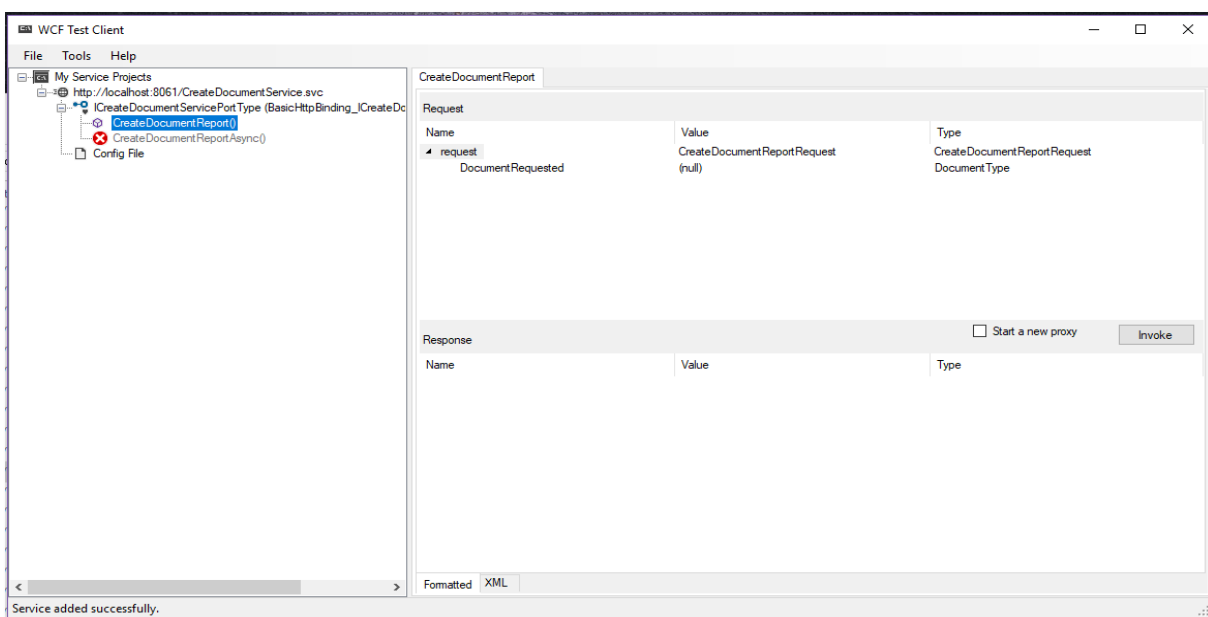


Figure 28 - Created service

- Provide the values for document type and 'invoke' the request.

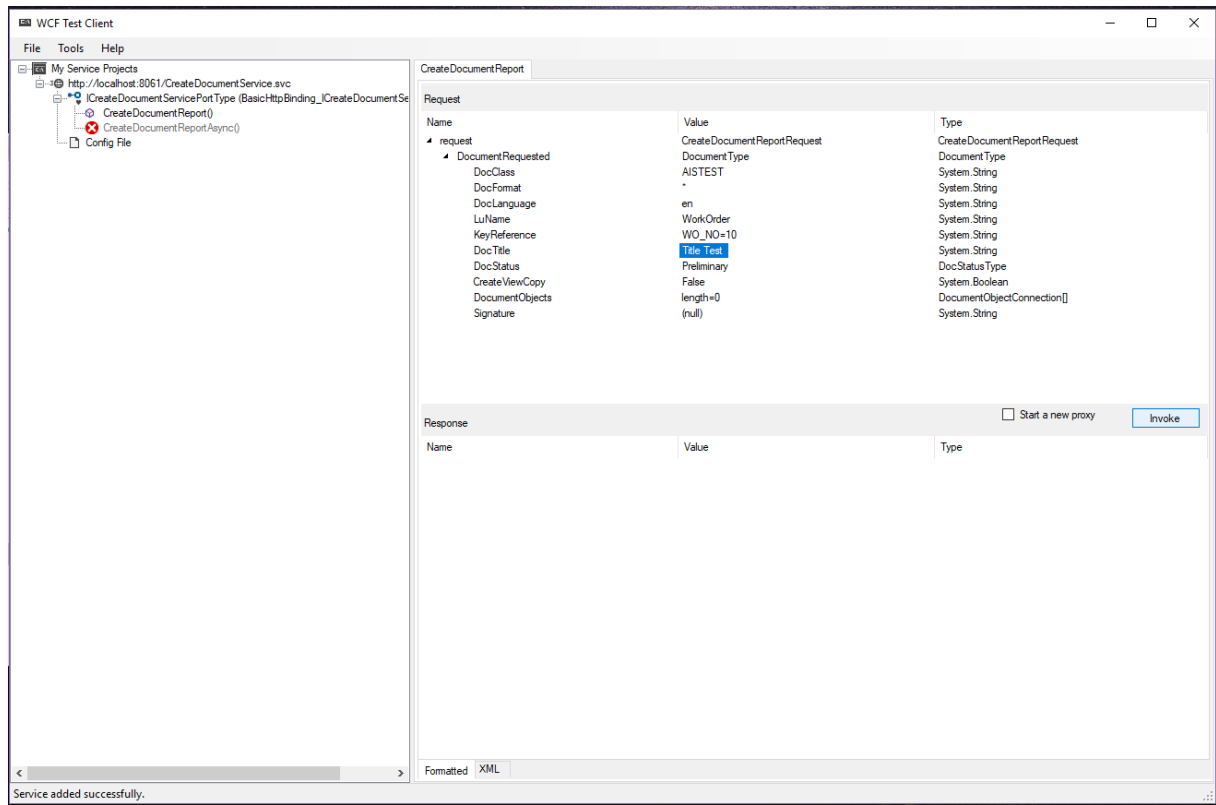


Figure 29 - Invoke Request

- Result will be displayed with the information of checked-in details.

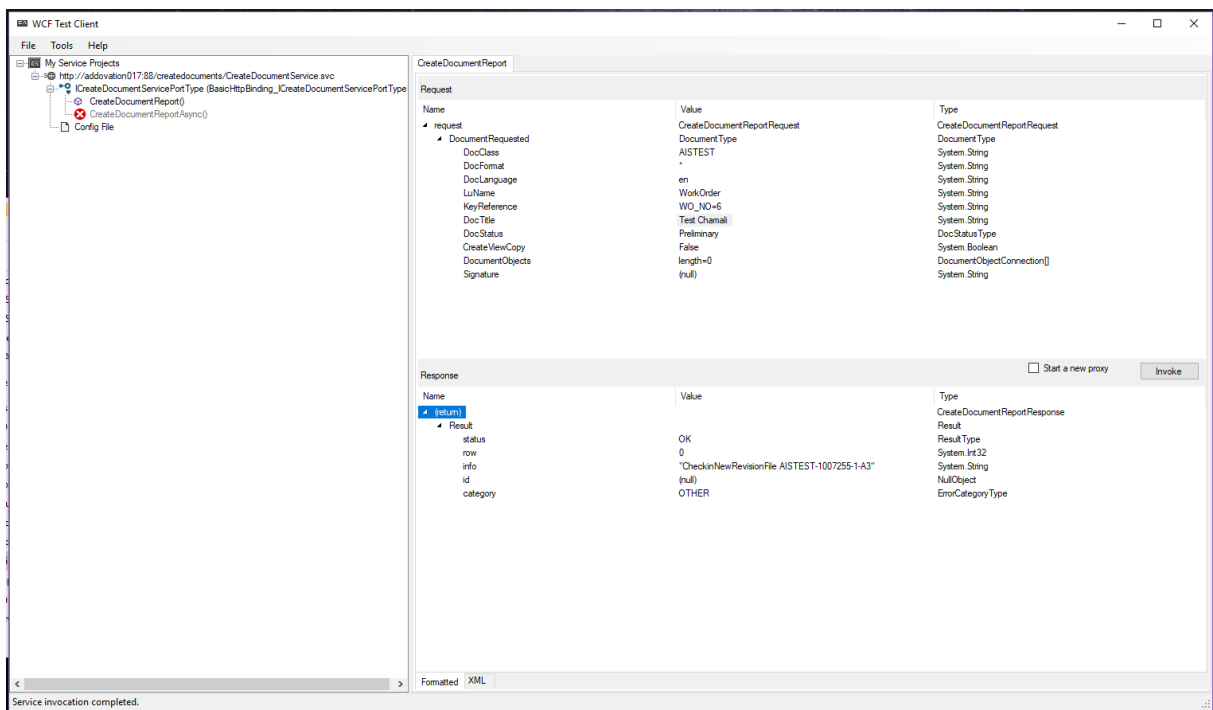
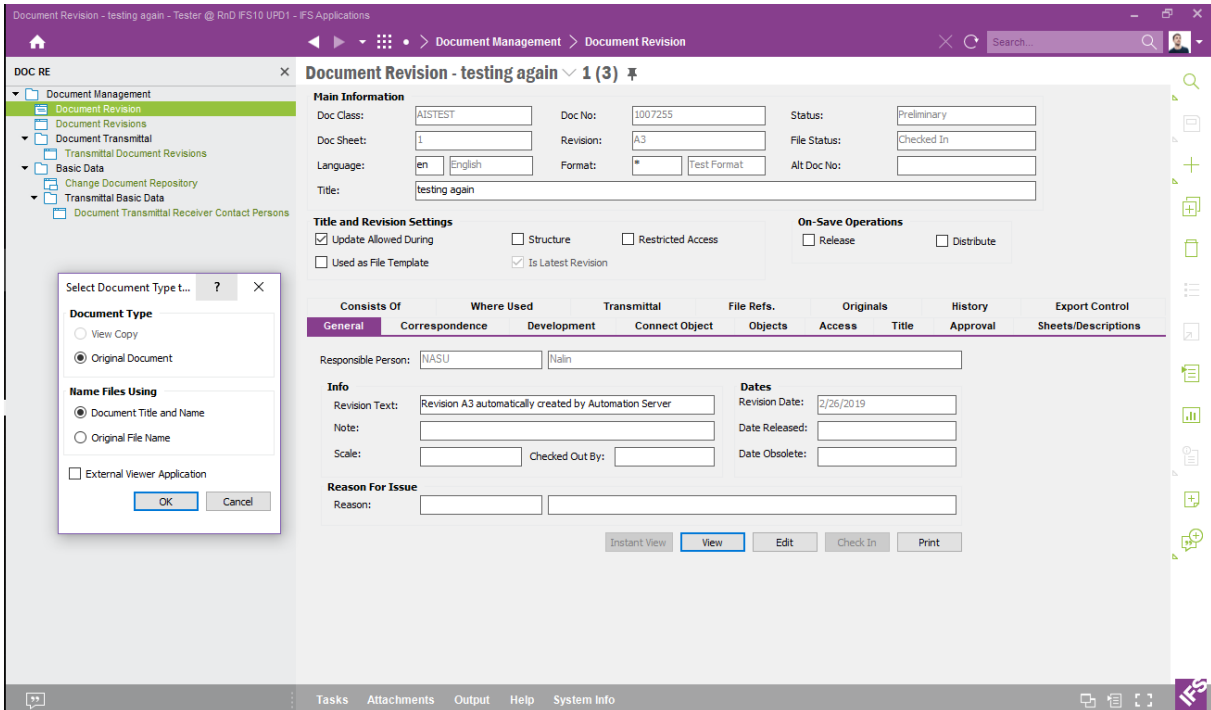


Figure 30 - Response

- Created document revision in IFS Applications.



Document Revision - testing again - Tester @ RnD IFS10 UPD1 - IFS Applications

Document Management > Document Revision

DOC RE

Document Management

Document Revisions

Document Transmittal

Transmittal Document Revisions

Basic Data

Change Document Repository

Transmittal Basic Data

Document Transmittal Receiver Contact Persons

Document Revision - testing again 1 (3)

Main Information

Doc Class: AISTEST Doc No: 1007255 Status: Preliminary

Doc Sheet: 1 Revision: A3 File Status: Checked In

Language: en English Format: Test Format Alt Doc No:

Title: testing again

Title and Revision Settings

☒ Update Allowed During ☐ Structure ☐ Restricted Access

☐ Used as File Template ☒ Is Latest Revision

On-Save Operations

☐ Release ☐ Distribute

Consists Of

General	Correspondence	Development	Connect Object	File Refs.	Originals	History	Export Control
General				Objects	Access	Title	Approval

Responsible Person: NASU Main

Info

Revision Text: Revision A3 automatically created by Automation Server

Note:

Scale: Checked Out By:

Dates

Revision Date: 2/26/2019

Date Released:

Date Obsolete:

Reason For Issue

Reason:

Instant View View Edit Check In Print

Tasks Attachments Output Help System Info

Select Document Type t...

Document Type

☐ View Copy

☒ Original Document

Name Files Using

☒ Document Title and Name

☐ Original File Name

☐ External Viewer Application

OK Cancel

Figure 31 - Created Document Revision

1.15 IFS Applications – Mapping data to templates

For templates that is used in IFS Applications, Addovation Office Automation Suite is required to create templates. Information on how to create these templates exists in the user guide for Office Automation Suite.

1.16 IFS Cloud – Mapping data to templates

Mapping data to templates for IFS Cloud is different and does not require Addovation Office Automation Suite to be installed. The mapping consists of basically 3 parts:

- The Microsoft Word document that will be used as a template.
- EDM basic data in IFS Cloud that defines the document class, format size and language to be used for the template.
- A newly introduced mapping file between the template and data in IFS Cloud.
 - A default mapping file will be generated the first-time service is being used and then throw an error.

For IFS Cloud, there are mainly 2 types of data that will be injected into a Word document:

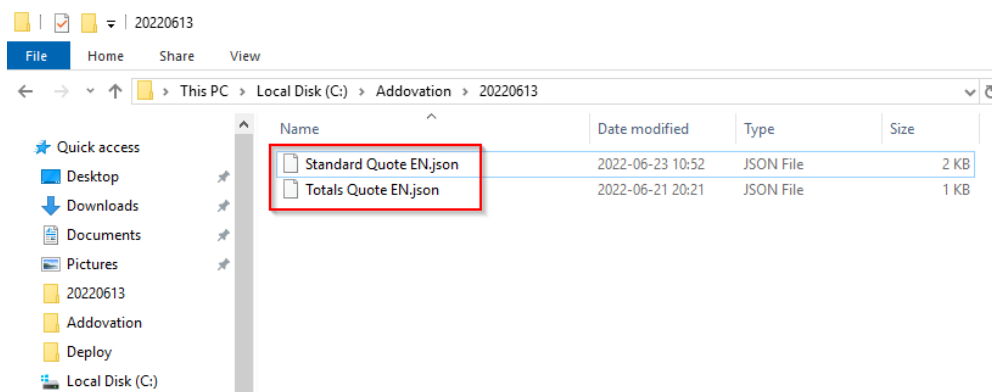
1. Document properties.
 - a. Properties are singular information such as details for a Customer Order.
 - b. Properties can be used as variables for quick report tables.
 - c. Additional singular document properties can also use quick reports.
2. Tables.
 - a. Tables are multiple rows of information such Customer Order Line.
 - b. Data source is quick reports.

For document properties, all names must be unique and correspond with what IFS Cloud delivers. For tables, the content control ID must be unique.

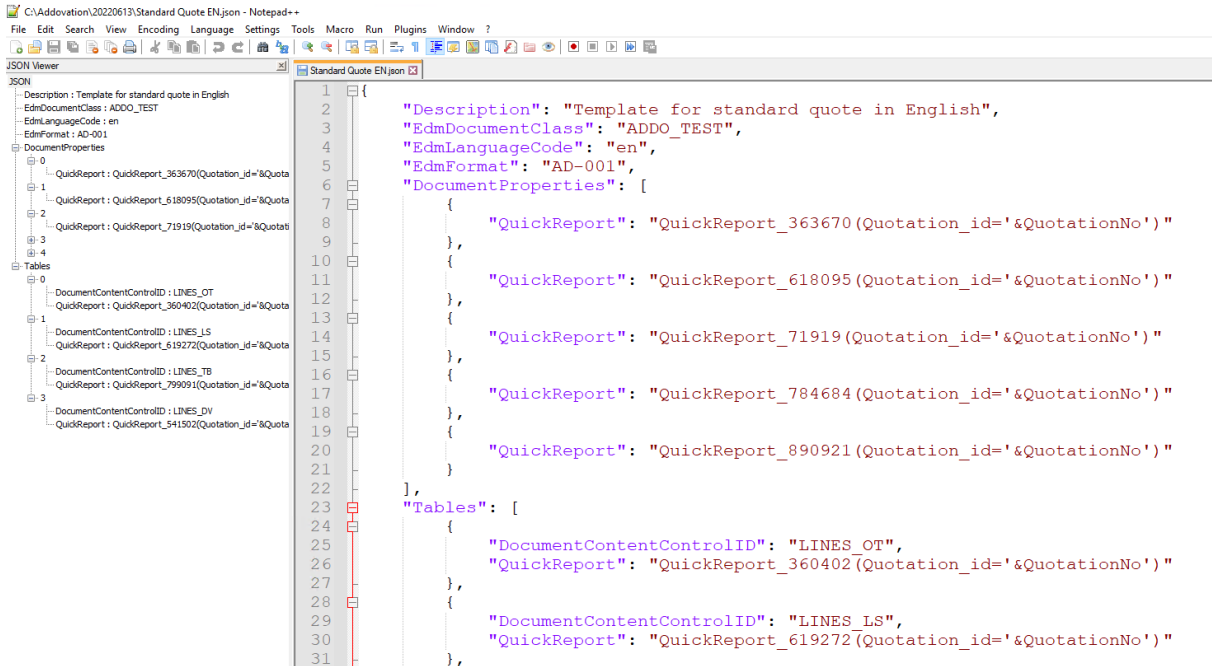
1.16.1 Mapping file explained

The newly introduced mapping file(s) is JSON file(s) that maps data between an EDM template and IFS Cloud data. Upon first time use of the service, a single default JSON file be generated must be modified using some appropriate editor such as notepad++. The location of these folders is being set in the web.config.

The names of the files do not matter, as long as they end with “.json” and the file can be deserialized into the used format. Good practice is to keep 1 file per template:



When it comes to editing these files, it might look like this:



```

1 {
2   "Description": "Template for standard quote in English",
3   "EdmDocumentClass": "ADDO_TEST",
4   "EdmLanguageCode": "en",
5   "EdmFormat": "AD-001",
6   "DocumentProperties": [
7     {
8       "QuickReport": "QuickReport_363670(Quotation_id='&QuotationNo')"
9     },
10    {
11      "QuickReport": "QuickReport_618095(Quotation_id='&QuotationNo')"
12    },
13    {
14      "QuickReport": "QuickReport_71919(Quotation_id='&QuotationNo')"
15    },
16    {
17      "QuickReport": "QuickReport_784684(Quotation_id='&QuotationNo')"
18    },
19    {
20      "QuickReport": "QuickReport_890921(Quotation_id='&QuotationNo')"
21    }
22  ],
23  "Tables": [
24    {
25      "DocumentContentControlID": "LINES_OT",
26      "QuickReport": "QuickReport_360402(Quotation_id='&QuotationNo')"
27    },
28    {
29      "DocumentContentControlID": "LINES_LS",
30      "QuickReport": "QuickReport_619272(Quotation_id='&QuotationNo')"
31    }
32  ]
33 }
  
```

As we can see here, both singular document properties and table information has been mapped into a given EDM template. Available variables are being mapped by using the &-character in front of the property name. Then they will be replaced in the request. Variables can also be hardcoded here if needed.

Description	Some simple description to make it easier for maintenance.
EdmDocumentClass	The EDM template document class as defined in IFS Cloud.
EdmLanguageCode	The EDM template language code as defined in IFS Cloud.
EdmFormat	The EDM template format size as defined in IFS Cloud.
DocumentProperties	Additional quick reports that will fetched as singular document properties.
Tables	The Microsoft Word content control ID that shall be mapped against a quick report.

These files (and the template itself) are being cached on the server so it is vital to clear the IIS cache before testing any changes in the mapping or the template. Please also note that all information and mapping has to be precise (a character wrong will cause failure to merge accurate information).

1.16.2 Mapping singular document properties – standard

For singular document properties, there are two types of default entity sets that is being injected into the document automatically:

- All fields from entity set DocIssue.
- All fields from the current logical unit default entity set.

DocIssue fields

What fields that is being returned depends on the customer installation (since it can be configured or customized). The RESTful invoke is being done against: DocumentRevisionHandling.svc/DocIssueSet.

One can use either Postman or IFS API Explorer to see exactly what fields that is being returned in your installation / IFS Cloud instance:

Get entities from DocIssueset

GET /DocIssueset

	AUTHORIZATIONS:	OpenAPI and basicAuth
Search...	QUERY PARAMETERS	
NewRevInfoSet		
NewSheetGeneralInfoForVirtualSet		
NewSheetDocReferenceObjectSet		
FileTemplateSet		
DocClassLowQuerySet		
SelectDocSet		
ParentDocSet		
ApprovalRoutingSet		
ConnectObjHeadVirtualSet		
ReleasedDocHeads		
DocmanExportControlSet		
ParentDocumentSet		
AddDocHeadVirtualSet		
DocTransmittalIssueDocume...		
PdfArchiveSet		
EdmFileSet		
DocIssueset		

Response samples

Content type: application/json

```
{
  "- value": [
    {
      "DocDate": "2019-10-10",
      "DocType": "2019-10-10",
      "DocStatus": "2019-10-10",
      "DocTitle": "2019-10-10",
      "DocDescription": "2019-10-10",
      "DocAuthor": "2019-10-10",
      "DocCreated": "2019-10-10",
      "DocModified": "2019-10-10",
      "DocDeleted": "2019-10-10",
      "DocArchived": "2019-10-10",
      "DocReleased": "2019-10-10",
      "DocClass": "2019-10-10",
      "DocSubClass": "2019-10-10",
      "DocParent": "2019-10-10",
      "DocChild": "2019-10-10",
      "DocSibling": "2019-10-10",
      "DocGrandchild": "2019-10-10",
      "DocGrandparent": "2019-10-10",
      "DocAncestor": "2019-10-10",
      "DocDescendant": "2019-10-10",
      "DocCollaborator": "2019-10-10",
      "DocReviewer": "2019-10-10",
      "DocApprover": "2019-10-10",
      "DocSigner": "2019-10-10",
      "DocOwner": "2019-10-10",
      "DocManager": "2019-10-10",
      "DocAdmin": "2019-10-10",
      "DocUser": "2019-10-10",
      "DocGuest": "2019-10-10",
      "DocAnonymous": "2019-10-10",
      "DocAuthenticated": "2019-10-10",
      "DocUnauthenticated": "2019-10-10",
      "DocEncrypted": "2019-10-10",
      "DocDecrypted": "2019-10-10",
      "DocCompressed": "2019-10-10",
      "DocDecompressed": "2019-10-10",
      "DocZipped": "2019-10-10",
      "DocUnzipped": "2019-10-10",
      "DocPackaged": "2019-10-10",
      "DocUnpackaged": "2019-10-10",
      "DocSigned": "2019-10-10",
      "DocUnsigned": "2019-10-10",
      "DocSealed": "2019-10-10",
      "DocUnsealed": "2019-10-10",
      "DocLocked": "2019-10-10",
      "DocUnlocked": "2019-10-10",
      "DocReadOnly": "2019-10-10",
      "DocWritable": "2019-10-10",
      "DocExecutable": "2019-10-10",
      "DocNonExecutable": "2019-10-10",
      "DocRunnable": "2019-10-10",
      "DocNotRunnable": "2019-10-10",
      "DocStandalone": "2019-10-10",
      "DocDependent": "2019-10-10",
      "DocIndependent": "2019-10-10",
      "DocRelated": "2019-10-10",
      "DocUnrelated": "2019-10-10",
      "DocAssociated": "2019-10-10",
      "DocDisassociated": "2019-10-10",
      "DocConnected": "2019-10-10",
      "DocDisconnected": "2019-10-10",
      "DocLinked": "2019-10-10",
      "DocUnlinked": "2019-10-10",
      "DocBound": "2019-10-10",
      "DocUnbound": "2019-10-10",
      "DocAttached": "2019-10-10",
      "DocDetached": "2019-10-10",
      "DocEmbedded": "2019-10-10",
      "DocNotEmbedded": "2019-10-10",
      "DocInlined": "2019-10-10",
      "DocOutlined": "2019-10-10",
      "DocCollapsed": "2019-10-10",
      "DocExpanded": "2019-10-10",
      "DocHidden": "2019-10-10",
      "DocVisible": "2019-10-10",
      "DocShown": "2019-10-10",
      "DocConcealed": "2019-10-10",
      "DocRevealed": "2019-10-10",
      "DocObscured": "2019-10-10",
      "DocUnobscured": "2019-10-10",
      "DocForgotten": "2019-10-10",
      "DocRemembered": "2019-10-10",
      "DocLost": "2019-10-10",
      "DocFound": "2019-10-10",
      "DocMissing": "2019-10-10",
      "DocPresent": "2019-10-10",
      "DocAbsent": "2019-10-10",
      "DocAvailable": "2019-10-10",
      "DocUnavailable": "2019-10-10",
      "DocAccessible": "2019-10-10",
      "DocInaccessible": "2019-10-10",
      "DocReachable": "2019-10-10",
      "DocUnreachable": "2019-10-10",
      "DocTraversable": "2019-10-10",
      "DocNotTraversable": "2019-10-10",
      "DocNavigable": "2019-10-10",
      "DocNonNavigable": "2019-10-10",
      "DocQueryable": "2019-10-10",
      "DocNotQueryable": "2019-10-10",
      "DocSearchable": "2019-10-10",
      "DocNotSearchable": "2019-10-10",
      "DocIndexable": "2019-10-10",
      "DocNotIndexable": "2019-10-10",
      "DocSortable": "2019-10-10",
      "DocNotSortable": "2019-10-10",
      "DocFilterable": "2019-10-10",
      "DocNotFilterable": "2019-10-10",
      "DocGroupable": "2019-10-10",
      "DocNotGroupable": "2019-10-10",
      "DocCategorizable": "2019-10-10",
      "DocNotCategorizable": "2019-10-10",
      "DocTaggable": "2019-10-10",
      "DocNotTaggable": "2019-10-10",
      "DocLabelable": "2019-10-10",
      "DocNotLabelable": "2019-10-10",
      "DocMarkable": "2019-10-10",
      "DocNotMarkable": "2019-10-10",
      "DocIdentifiable": "2019-10-10",
      "DocNotIdentifiable": "2019-10-10",
      "DocNameable": "2019-10-10",
      "DocNotNameable": "2019-10-10",
      "DocDescriptible": "2019-10-10",
      "DocNotDescriptible": "2019-10-10",
      "DocCharacterizable": "2019-10-10",
      "DocNotCharacterizable": "2019-10-10",
      "DocQualifiable": "2019-10-10",
      "DocNotQualifiable": "2019-10-10",
      "DocMeasurable": "2019-10-10",
      "DocNotMeasurable": "2019-10-10",
      "DocQuantifiable": "2019-10-10",
      "DocNotQuantifiable": "2019-10-10",
      "DocCountable": "2019-10-10",
      "DocNotCountable": "2019-10-10",
      "DocNumerical": "2019-10-10",
      "DocNotNumerical": "2019-10-10",
      "DocAlgebraic": "2019-10-10",
      "DocNotAlgebraic": "2019-10-10",
      "DocGeometric": "2019-10-10",
      "DocNotGeometric": "2019-10-10",
      "DocTopological": "2019-10-10",
      "DocNotTopological": "2019-10-10",
      "DocAnalytical": "2019-10-10",
      "DocNotAnalytical": "2019-10-10",
      "DocLogical": "2019-10-10",
      "DocNotLogical": "2019-10-10",
      "DocMathematical": "2019-10-10",
      "DocNotMathematical": "2019-10-10",
      "DocPhysical": "2019-10-10",
      "DocNotPhysical": "2019-10-10",
      "DocChemical": "2019-10-10",
      "DocNotChemical": "2019-10-10",
      "DocBiological": "2019-10-10",
      "DocNotBiological": "2019-10-10",
      "DocEnvironmental": "2019-10-10",
      "DocNotEnvironmental": "2019-10-10",
      "DocSocial": "2019-10-10",
      "DocNotSocial": "2019-10-10",
      "DocPolitical": "2019-10-10",
      "DocNotPolitical": "2019-10-10",
      "DocEconomic": "2019-10-10",
      "DocNotEconomic": "2019-10-10",
      "DocLegal": "2019-10-10",
      "DocNotLegal": "2019-10-10",
      "DocMedical": "2019-10-10",
      "DocNotMedical": "2019-10-10",
      "DocEducational": "2019-10-10",
      "DocNotEducational": "2019-10-10",
      "DocReligious": "2019-10-10",
      "DocNotReligious": "2019-10-10",
      "DocArtistic": "2019-10-10",
      "DocNotArtistic": "2019-10-10",
      "DocScientific": "2019-10-10",
      "DocNotScientific": "2019-10-10",
      "DocHistorical": "2019-10-10",
      "DocNotHistorical": "2019-10-10",
      "DocLiterary": "2019-10-10",
      "DocNotLiterary": "2019-10-10",
      "DocLinguistic": "2019-10-10",
      "DocNotLinguistic": "2019-10-10",
      "DocPhilosophical": "2019-10-10",
      "DocNotPhilosophical": "2019-10-10",
      "DocPsychological": "2019-10-10",
      "DocNotPsychological": "2019-10-10",
      "DocPhysiological": "2019-10-10",
      "DocNotPhysiological": "2019-10-10",
     
```

In order to use any of these fields in a Word template, simply create a new document property for the template. It is important though that the name of the property corresponds with what IFS Cloud delivers.

In order to create a new document property in the template, go to the File tab and Info and then Properties and Advanced Properties as following:

Document1 - Word

Info

Home

Document Management

New

Open

Info

Save

Save As

Print

Share

Export

Transform

Close

Protect Document
Control what types of changes people can make to this document.

Inspect Document
Before publishing this file, be aware that it contains:

- Document properties and author's name

Version History
View and restore previous versions.

Manage Document
There are no unsaved changes.

Properties ▾

Advanced Properties
See more document properties

Total Editing Time 0 Minutes

Title Add a title

Tags Add a tag

Comments Add comments

Related Dates

Last Modified

Created Today, 08:08

Last Printed

Related People

Author

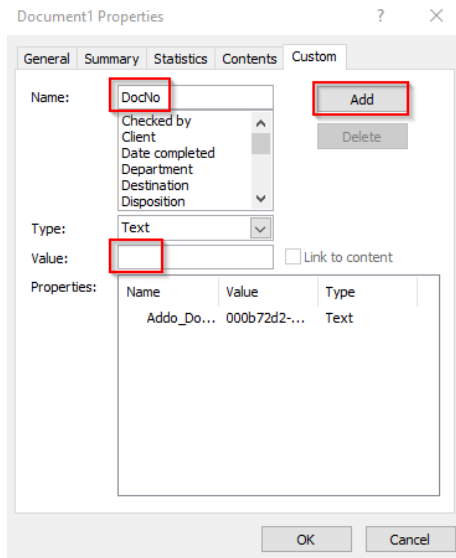
AI Andreas Isengaard

Add an author

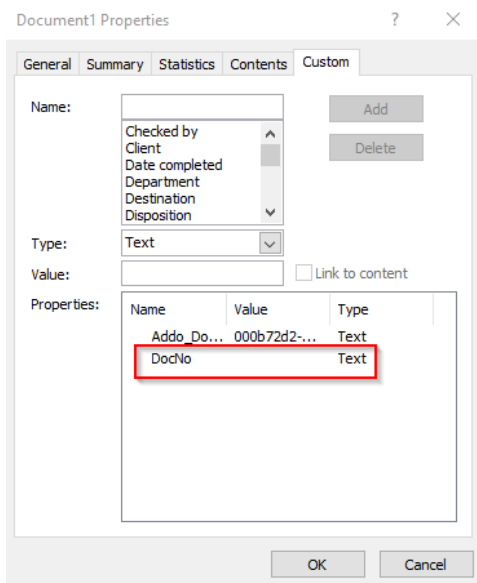
Last Modified By Not saved yet

[Show All Properties](#)

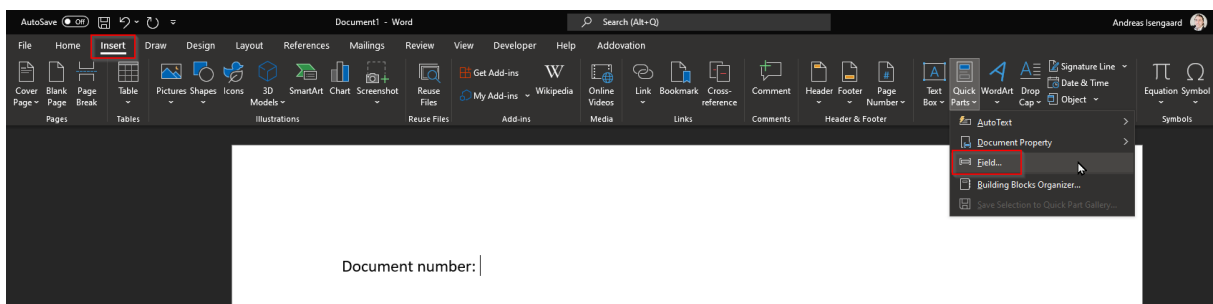
In our example, let's add DocNo so we enter the exact name as a new document property:



Add a space or something else as default value (because Microsoft Word has a limitation that requires always a value) and then press Add. Now you should have something like this:

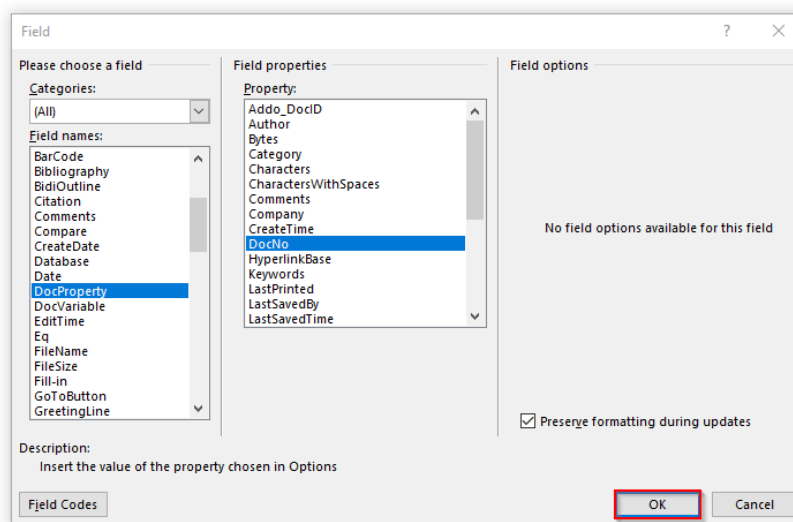


In order to use this property now in the document, click OK and head back to Microsoft Word. Go to the Insert tab and select Quick Parts and Field as such:

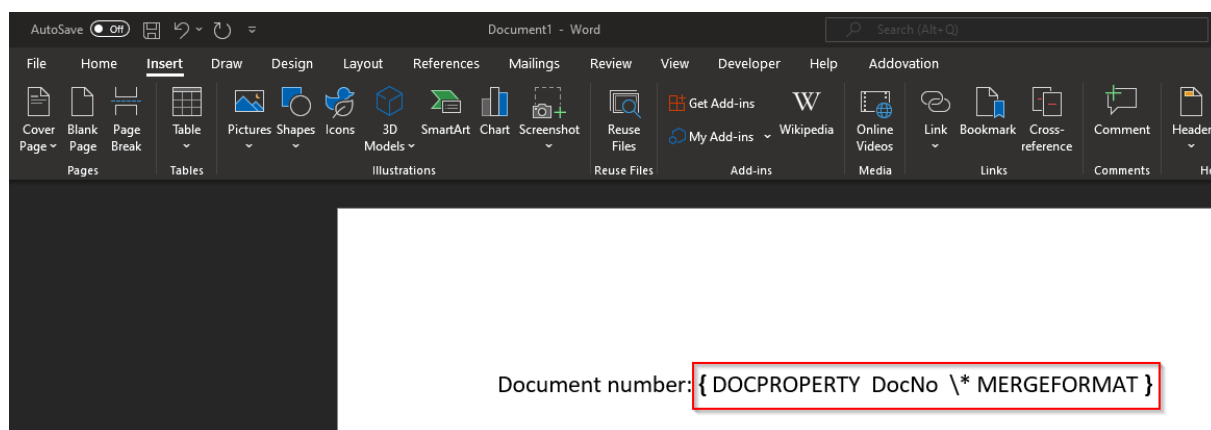


Scroll down to DocProperty field names, select the newly created DocNo as field property and then press OK:

Document number:



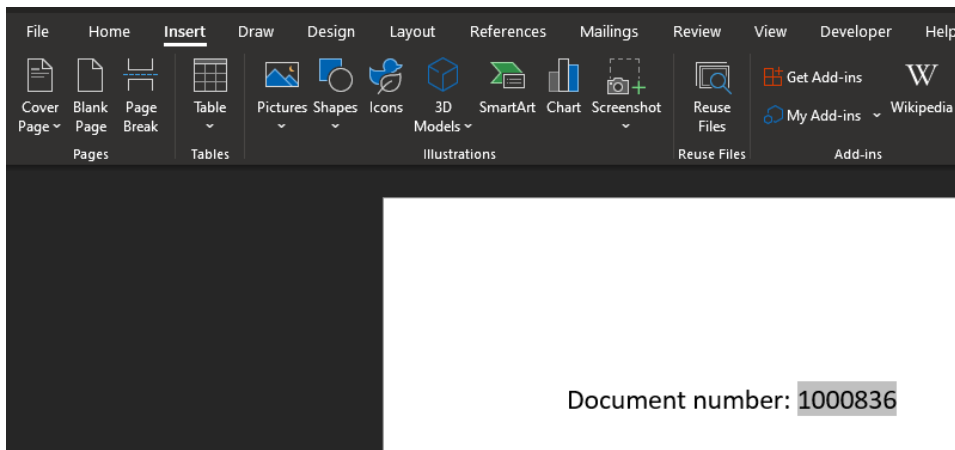
Now you should have something like this in the Word template:



By using Alt + F9 one can toggle showing property placeholders on and off in Microsoft Word. Now, last steps are to:

1. Update EDM template in IFS Cloud.
2. Refresh the application in IIS or simply reset whole IIS.
3. Test the new template by running the request as explained previously using WCF Test Client.

Then one should see something like this:



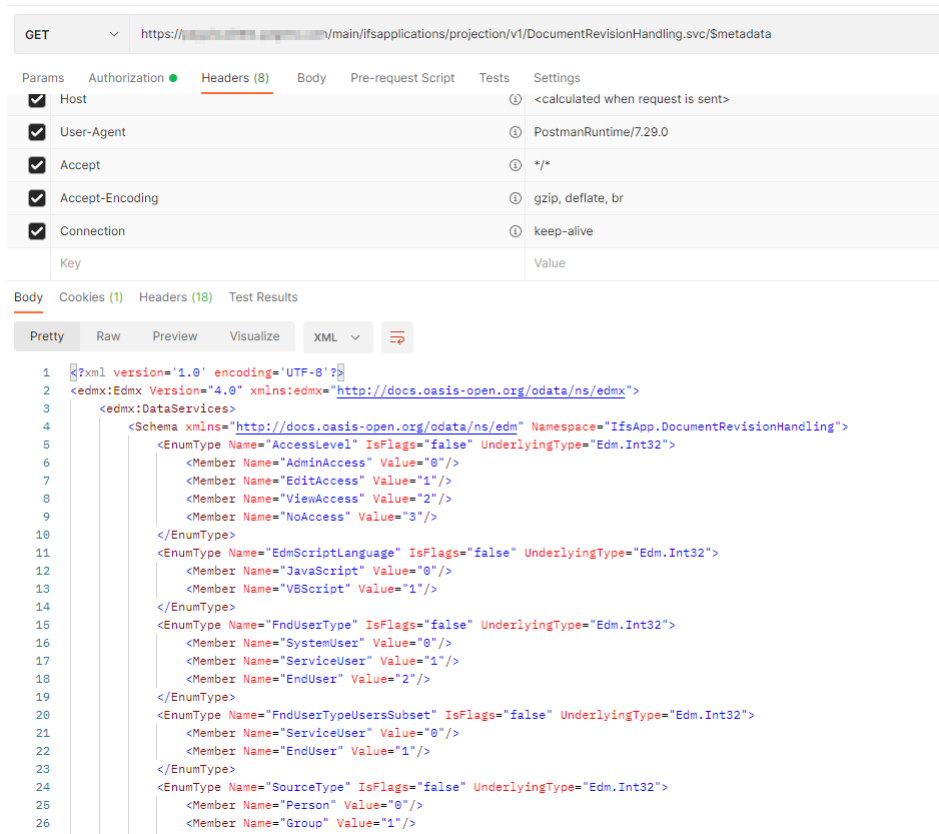
Now we see that the document property was injected into the document upon creation in IFS Cloud.

Dynamic fields from the logical unit

In addition to DocIssue, fields from the object connection are also being added into the document. The service follows same logic as IFS Cloud when it comes to extracting information and fields. The source for information in IFS Cloud is DocumentRevisionHandling.svc/\$metadata. If the logical unit has been setup properly in object connections, there will be something called a dynamic reference in the metadata set:

[illegible]

If we are to connect against e.g. the logical unit `OrderQuotation`, we can see what fields that will be returned from IFS Cloud using Postman:



```

1 <?xml version='1.0' encoding='UTF-8'?>
2 <edmx:Edmx Version="4.0" xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx">
3   <edmx:DataServices>
4     <Schema xmlns="http://docs.oasis-open.org/odata/ns/edm" Namespace="IfsApp.DocumentRevisionHandling">
5       <EnumType Name="AccessLevel" IsFlags="false" UnderlyingType="Edm.Int32">
6         <Member Name="AdminAccess" Value="0"/>
7         <Member Name="EditAccess" Value="1"/>
8         <Member Name="ViewAccess" Value="2"/>
9         <Member Name="NoAccess" Value="3"/>
10      </EnumType>
11      <EnumType Name="EdmScriptLanguage" IsFlags="false" UnderlyingType="Edm.Int32">
12        <Member Name="JavaScript" Value="0"/>
13        <Member Name="VBScript" Value="1"/>
14      </EnumType>
15      <EnumType Name="FndUserType" IsFlags="false" UnderlyingType="Edm.Int32">
16        <Member Name="SystemUser" Value="0"/>
17        <Member Name="ServiceUser" Value="1"/>
18        <Member Name="EndUser" Value="2"/>
19      </EnumType>
20      <EnumType Name="FndUserTypeUsersSubset" IsFlags="false" UnderlyingType="Edm.Int32">
21        <Member Name="ServiceUser" Value="0"/>
22        <Member Name="EndUser" Value="1"/>
23      </EnumType>
24      <EnumType Name="SourceType" IsFlags="false" UnderlyingType="Edm.Int32">
25        <Member Name="Person" Value="0"/>
26        <Member Name="Group" Value="1"/>

```

After that, search for “DynLov_<LogicalUnit>” so what we are looking for then is “DynLov_OrderQuotation” in this case we have something like this:



```

6594 <EntityType Name="DynLov_OrderQuotation">
6595   <Key>
6596     <PropertyRef Name="QuotationNo"/>
6597   </Key>
6598   <Property Name="keyref" Type="Edm.String" MaxLength="4000"/>
6599   <Property Name="QuotationNo" Type="Edm.String" MaxLength="12"/>
6600   <Property Name="RequestReceivedDate" Type="Edm.Date"/>
6601   <Property Name="AnsweringDate" Type="Edm.Date"/>
6602   <Property Name="RevisionNo" Type="Edm.Decimal" Precision="130" Scale="130"/>
6603   <Property Name="ClosedStatus" Type="Edm.String" MaxLength="200"/>
6604   <Property Name="Printed" Type="Edm.String" MaxLength="200"/>
6605   <Property Name="JinsulInvoice" Type="Edm.String" MaxLength="200"/>
6606   <Property Name="ForwardAgentId" Type="Edm.String" MaxLength="20"/>
6607   <Property Name="FreightMapId" Type="Edm.String" MaxLength="15"/>
6608   <Property Name="ZoneId" Type="Edm.String" MaxLength="15"/>
6609   <Property Name="FreightPricelistNo" Type="Edm.String" MaxLength="10"/>
6610   <Property Name="ApplyFixDelivFreight" Type="Edm.String" MaxLength="200"/>
6611   <Property Name="FixDelivFreight" Type="Edm.Decimal" Precision="130" Scale="130"/>
6612   <Property Name="VendorNo" Type="Edm.String" MaxLength="20"/>
6613   <Property Name="FreeOfChgTaxPayParty" Type="Edm.String" MaxLength="200"/>
6614   <Property Name="CIntervalUnit" Type="Edm.String" MaxLength="200"/>
6615   <Property Name="CUnit" Type="Edm.String" MaxLength="200"/>
6616   <Property Name="CProjectId" Type="Edm.String" MaxLength="10"/>
6617   <Property Name="State" Type="Edm.String" MaxLength="20"/>
6618 </EntityType>

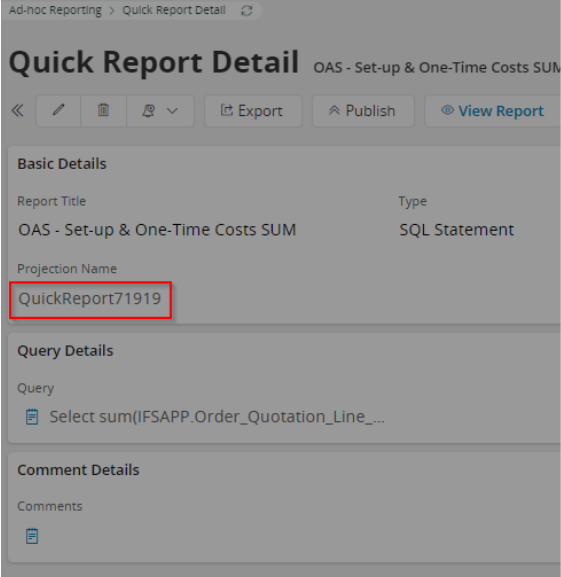
```

Here we see all the additional fields and properties that we can use as variables or directly in our Word template.

1.16.3 Mapping singular document properties – using quick reports

Sometimes, one needs to have additional document properties. This can be achieved by using quick reports with or without input variables.

In order to proceed make sure you have a quick report ready and uses any of these input variables and that the integration user has access to it. If we want to have this quick report mapped in our Word template, we note the projection name, the column name in the quick report result (here we see “SUM”) and the variable name (case sensitive):



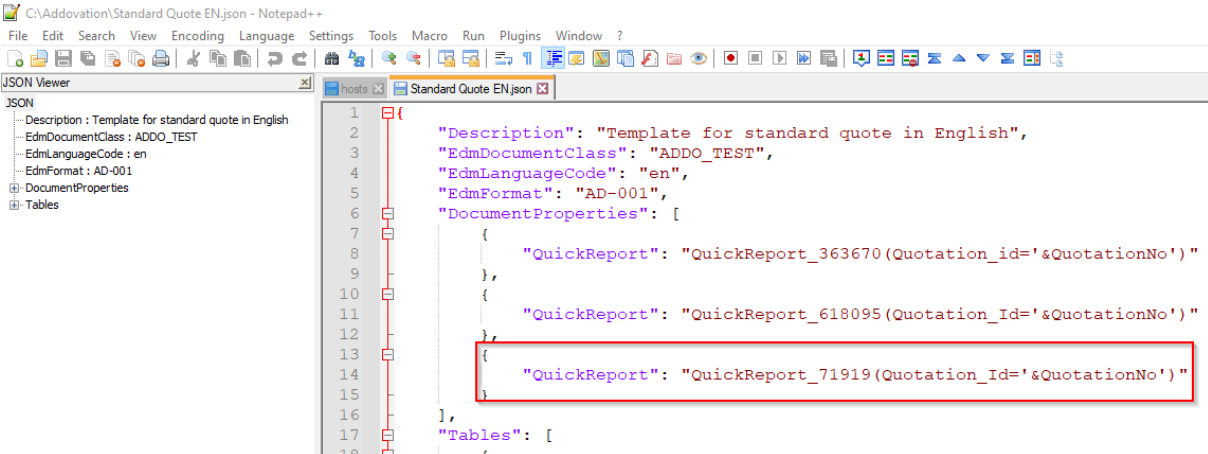
```

Select sum(IFSAPP.Order_Quotation_Line_API.Get_Sale_Price_Total(quotation_no,
line_no,
rel_no,
line_item_no) "SUM"
FROM IFSAPP.ORDER_QUOTATION_LINE T, IFSAPP.PART_CATALOG U
where t.catalog_no = u.part_no
AND c_product_type_id IN ('Setup')
AND t.quotations_id = '&Quotation_id'
                    
```

Make sure also that a Word document property exists in the template with the name “SUM”.

Next step is to add it to the mapping file. The mapping file exists locally on the server in where the AIS CreateDocumentService has been installed as previously explained. RDP to the server and start to edit the mapping file.

Add a new definition as following:

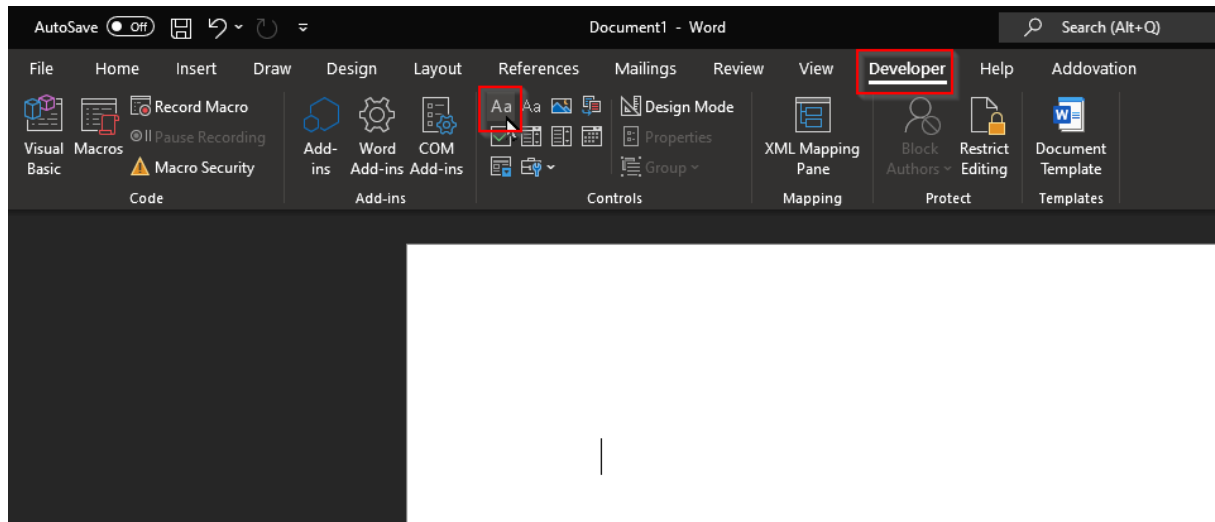


If multiple variables are being used, then use IFS Cloud debug console to see how several variables are being set in the request URL.

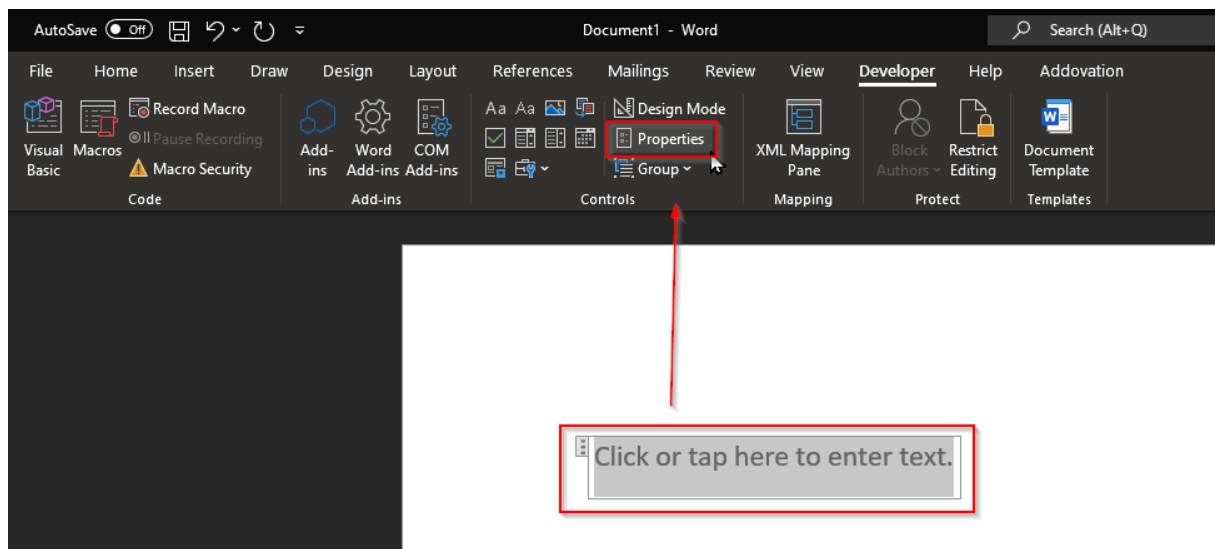
After editing has been done, then clear the IIS cache and re-test to see the result.

1.16.4 Mapping tables

Mapping of tables requires usage of rich text content controls in Microsoft Word. Continue by adding a new rich text control like so using the Developer tab (it is not enabled by default so one must enable it in Microsoft office):

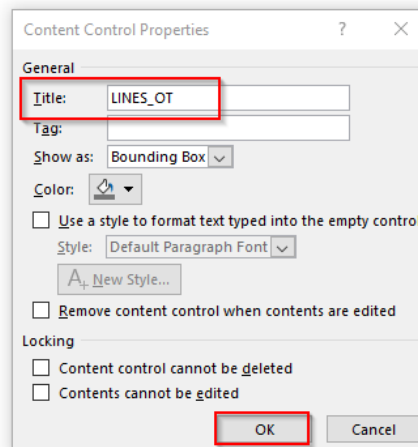


Then it will create a place holder for the table as such, after that click the Properties in order to give it a unique name that we will map against a quick report:



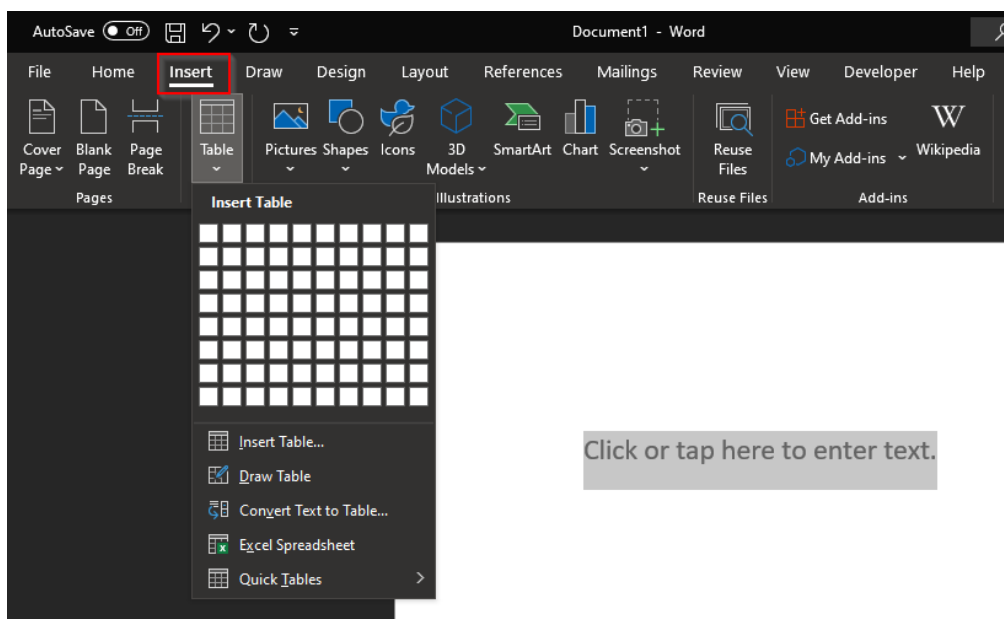
In our example, we will call it "LINES_OT" and press OK (it is crucial that we enter the name in the Title field):

Click or tap here to enter text.

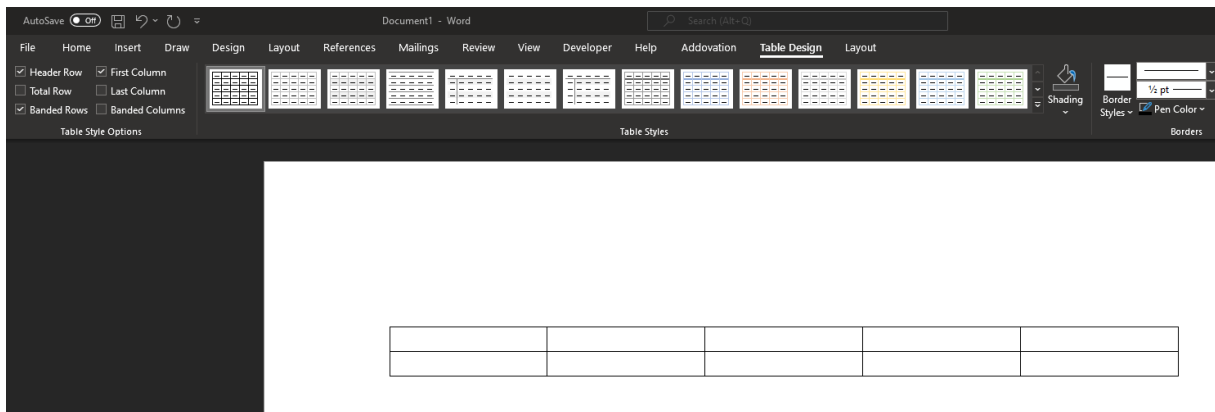


The 'Content Control Properties' dialog box is shown with the 'General' tab selected. The 'Title' field is highlighted with a red box and contains the text 'LINES_OT'. The 'Tag' field is empty. The 'Show as' dropdown is set to 'Bounding Box'. The 'Color' dropdown is set to a light blue color. The 'Use a style to format text typed into the empty control' checkbox is unchecked, and the 'Style' dropdown is set to 'Default Paragraph Font'. The 'Remove content control when contents are edited' checkbox is unchecked. The 'Locking' section has two unchecked checkboxes: 'Content control cannot be deleted' and 'Contents cannot be edited'. The 'OK' button is highlighted with a red box.

Next step is to add the table itself within the content control. Click on the Insert tab in Microsoft Word and select the number of columns you desire to have and always make it 2 rows (1 row for columns and 1 row for the data itself to be repeated in):



In our example, we want to have 5 columns from the quick report – so we simply add 5 columns with 2 rows:

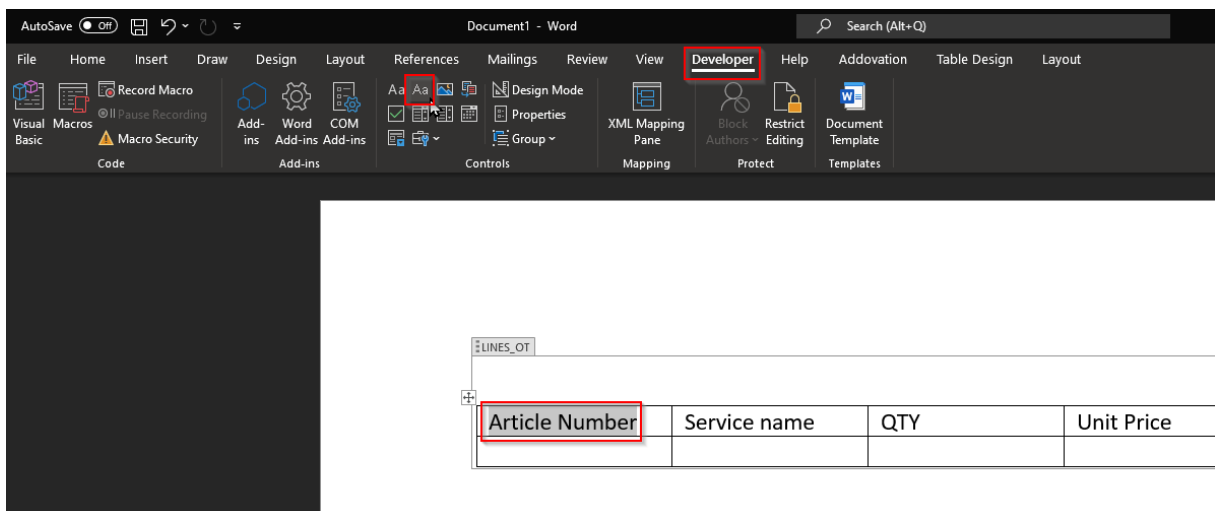


Now we add the different column names that will be mapped against the quick report. We will add a display value that can be anything and then convert it into a plain text content control that will be used for mapping.

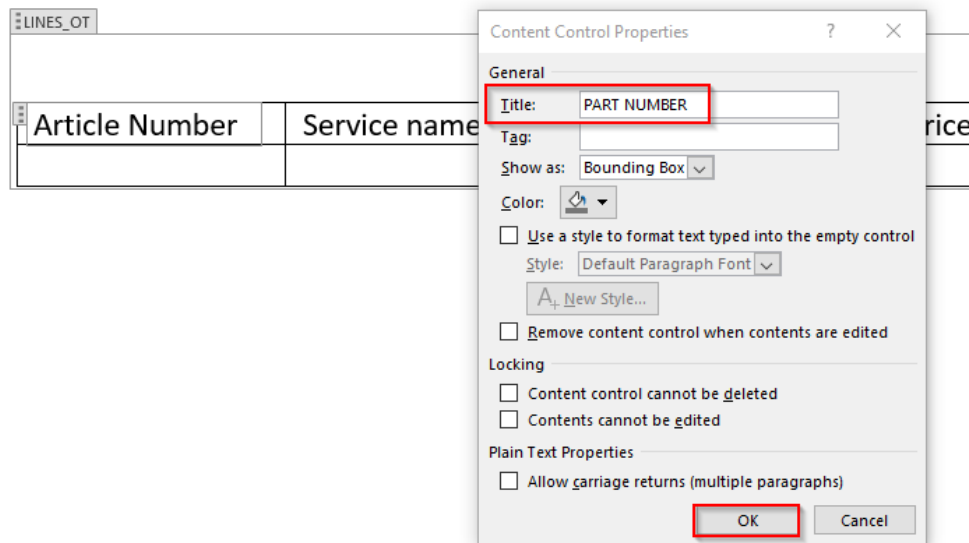
Continue to add all the display values as following, for instance:

Article Number	Service name	QTY	Unit Price	Total

Next step is to select the first header, in this case “Article Number” and convert it into a plain text content control:



After that, click on Properties and we will here map it against the column name in the quick report (in our example the column name in the quick report is “PART NUMBER”):



Next press OK and repeat the process with all column names. Last step we will format the table and make it look a bit prettier, for instance:

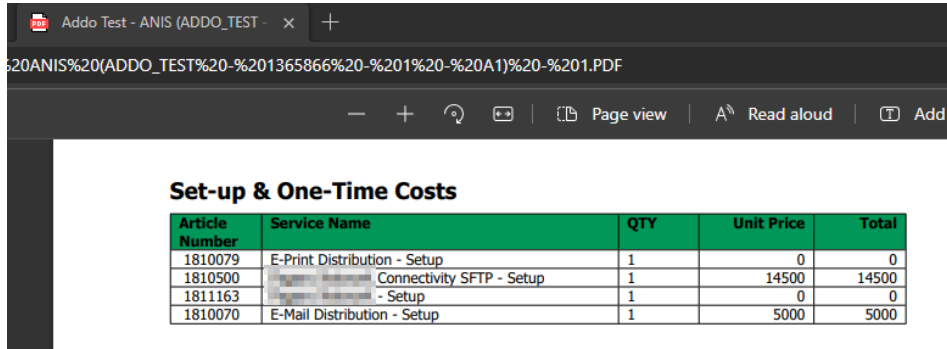
Article Number	Service Name	QTY	Unit Price	Total

Now, head to the mapping file and add the newly created content control:

```

1  {
2      "Description": "Template for standard quote in English",
3      "EdmDocumentClass": "ADDO_TEST",
4      "EdmLanguageCode": "en",
5      "EdmFormat": "AD-001",
6      "DocumentProperties": [
7          {
8              "QuickReport": "QuickReport_363670(Quotation_id='&QuotationNo') "
9          },
10         {
11             "QuickReport": "QuickReport_618095(Quotation_Id='&QuotationNo') "
12         },
13         {
14             "QuickReport": "QuickReport_71919(Quotation_Id='&QuotationNo') "
15         }
16     ],
17     "Tables": [
18         {
19             "DocumentContentControlID": "LINES_OT",
20             "QuickReport": "QuickReport_360402(Quotation_id='&QuotationNo') "
21         },
22     ]
23 }
```

Here we enter the word content control and it's title "LINES_OT" and then what quick report it should be mapped against and any eventual parameters and input variables. Save the file on the server and then refresh the cache. Check in all appended changes for the EDM template and test the template. If everything was done correct you should now see the results as:



The screenshot shows a web browser window with a tab titled 'Addo Test - ANIS (ADDO_TEST - x +'. The address bar shows a URL ending in '.PDF'. The browser interface includes buttons for 'Page view', 'Read aloud', and 'Add'. The main content area displays a table titled 'Set-up & One-Time Costs'.

Article Number	Service Name	QTY	Unit Price	Total
1810079	E-Print Distribution - Setup	1	0	0
1810500	Connectivity SFTP - Setup	1	14500	14500
1811163	- Setup	1	0	0
1810070	E-Mail Distribution - Setup	1	5000	5000

In order to delete the content control if no data has been found on need to set the following value in the mapping file:

```

17  },
18  },
19  {
20    "QuickReport": "QuickReport_890921(Quotation_id='&QuotationNo') "
21  },
22  ],
23  "Tables": [
24    {
25      "DocumentContentControlID": "LINES_OT",
26      "QuickReport": "QuickReport_360402(Quotation_id='&QuotationNo') "
27    },
28    {
29      "DocumentContentControlID": "LINES_LS",
30      "QuickReport": "QuickReport_619272(Quotation_id='&QuotationNo') "
31    },
32    {
33      "DocumentContentControlID": "LINES_TB",
34      "QuickReport": "QuickReport_799091(Quotation_id='&QuotationNo') "
35    },
36    {
37      "DocumentContentControlID": "LINES_TB2",
38      "QuickReport": "QuickReport_799091(Quotation_id='&QuotationNo') ",
39      "DeleteContentControlIfNoDataFound": true
40    },
41    {
42      "DocumentContentControlID": "LINES_DV",
43      "QuickReport": "QuickReport_541502(Quotation_id='&QuotationNo') "
44    }
45  ]
46  ]
47  }

```

This applies to scenarios in where want to have more dynamic tables.

2 Upgrade notes

2.1 Upgrading old versions to Automation Server 2.1. 33715 or higher from older versions.

While performing an upgrade, environments and license key should be added again, since previously configured environments won't be available during and after the upgrade.

3 Update AIS CreateDocumentService via zip file

RDP to the server: Ex: example.customerA.local.

Copy zip-file to desktop and unzip folder it.

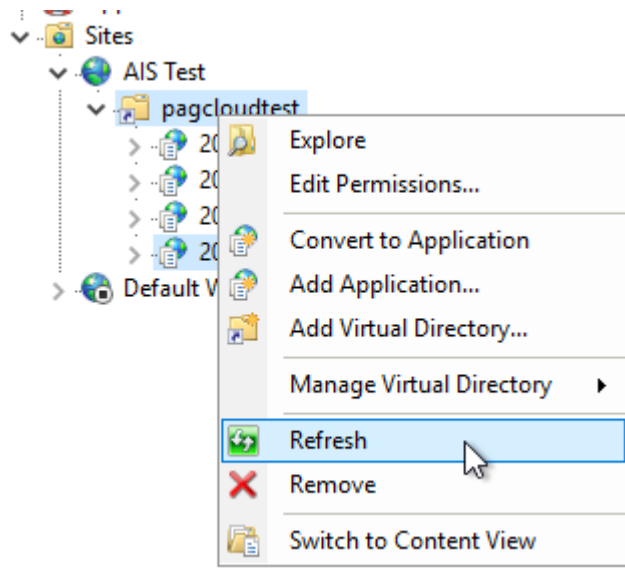
Go to the hosted site. Ex: C:\inetpub\wwwroot_ais\pagcloudtest.

Create a new folder with a suitable date in e.g. "20220630". The name will be used in URL to service.

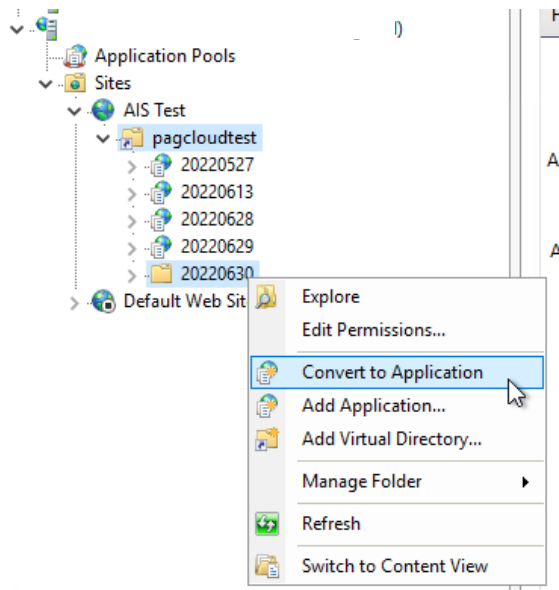
Copy all files from zip folder to root in newly created folder.

Copy web.config from previous deployment.

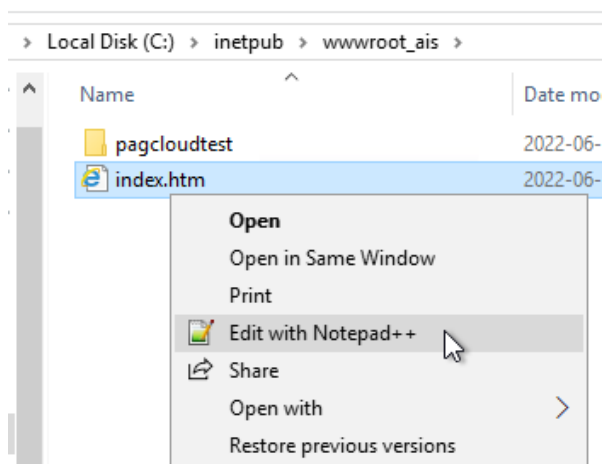
Open IIS manager and refresh AIS Test and virtual sub folder "pagcloudtest":



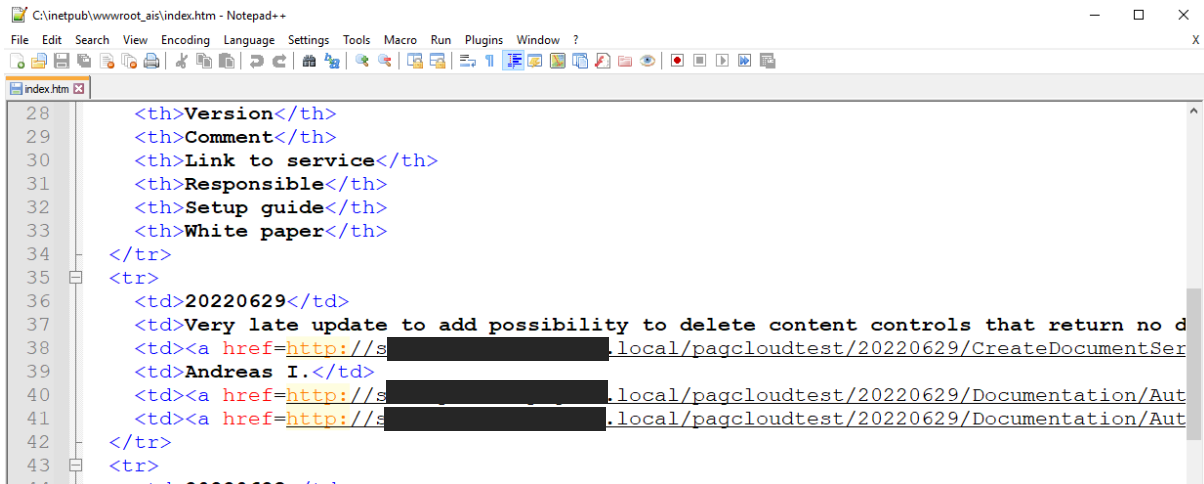
RMB on the new folder and click "Convert to Application" and accept default settings:



Optional: Edit index.htm file in order to add some comments and to keep history:



Add a new row and point to the new service:

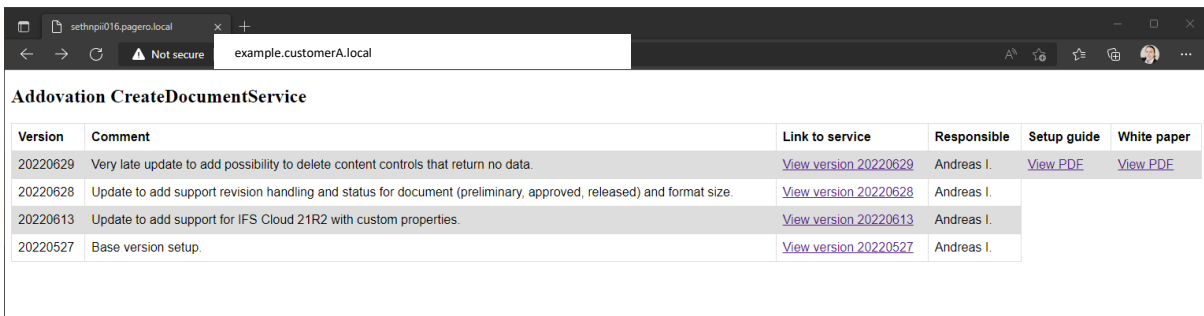


```

28 <th>Version</th>
29 <th>Comment</th>
30 <th>Link to service</th>
31 <th>Responsible</th>
32 <th>Setup guide</th>
33 <th>White paper</th>
34 </tr>
35 <tr>
36 <td>20220629</td>
37 <td>Very late update to add possibility to delete content controls that return no d
38 <td><a href=http://s[redacted].local/pagcloudtest/20220629/CreateDocumentSer
39 <td>Andreas I.</td>
40 <td><a href=http://s[redacted].local/pagcloudtest/20220629/Documentation/Aut
41 <td><a href=http://s[redacted].local/pagcloudtest/20220629/Documentation/Aut
42 </tr>
43 <tr>
44 <td>20220628</td>

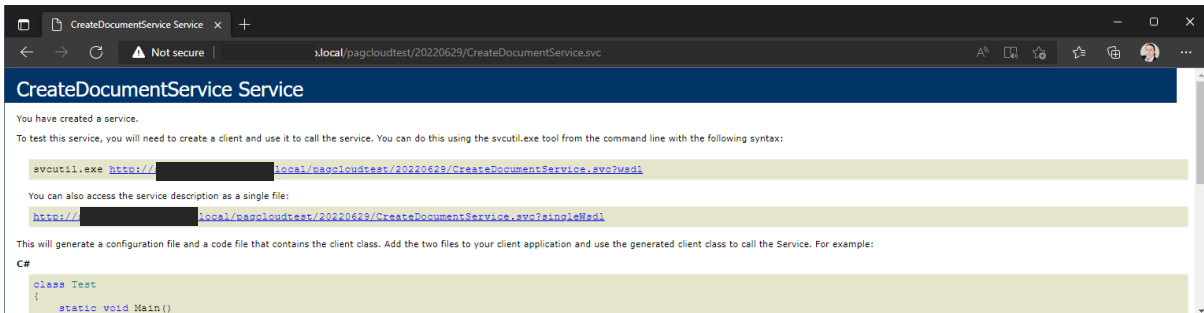
```

Open up the URL to the server in the browser against: example.customerA.local.



Version	Comment	Link to service	Responsible	Setup guide	White paper
20220629	Very late update to add possibility to delete content controls that return no data.	View version 20220629	Andreas I.	View PDF	View PDF
20220628	Update to add support revision handling and status for document (preliminary, approved, released) and format size.	View version 20220628	Andreas I.		
20220613	Update to add support for IFS Cloud 21R2 with custom properties.	View version 20220613	Andreas I.		
20220527	Base version setup.	View version 20220527	Andreas I.		

Click on the “View version yyyyMMdd” to make sure service is healthy:



CreateDocumentService Service

You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the command line with the following syntax:

```

svcutil.exe http://[redacted].local/pagcloudtest/20220629/CreateDocumentService.svc?wsdl

```

You can also access the service description as a single file:

```

http://[redacted].local/pagcloudtest/20220629/CreateDocumentService.svc?singleWsdl

```

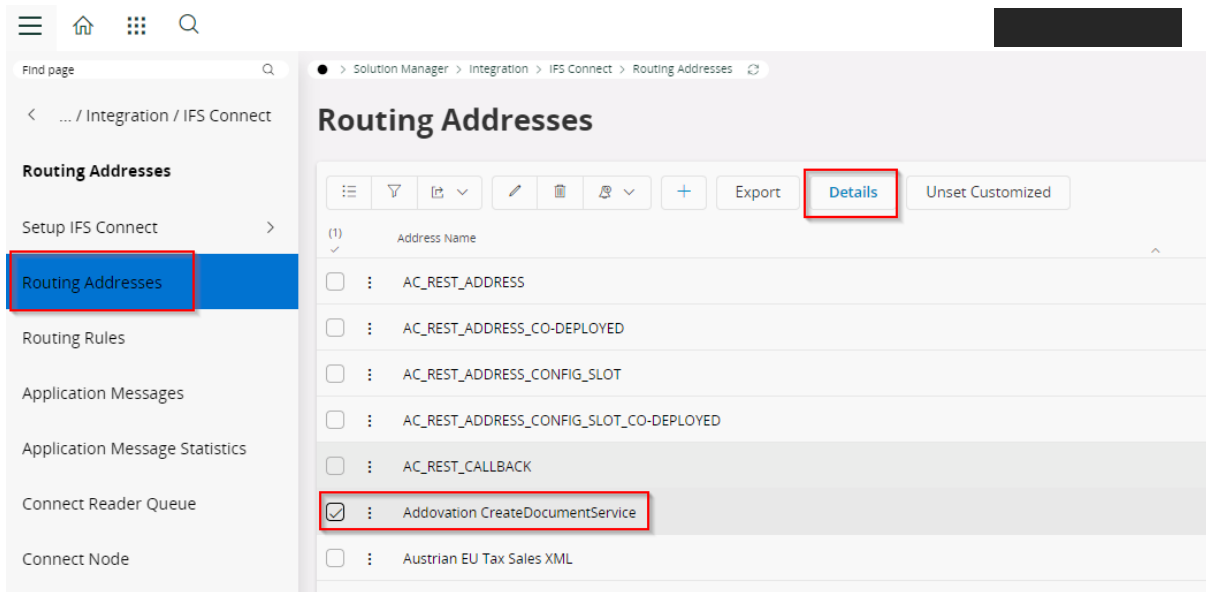
This will generate a configuration file and a code file that contains the client class. Add the two files to your client application and use the generated client class to call the Service. For example:

```

C#
class Test
{
    static void Main()
    {

```

In order to use the new version from IFS, head to routing address in IFS Connect, locate Addovation CreateDocumentService and click on details:

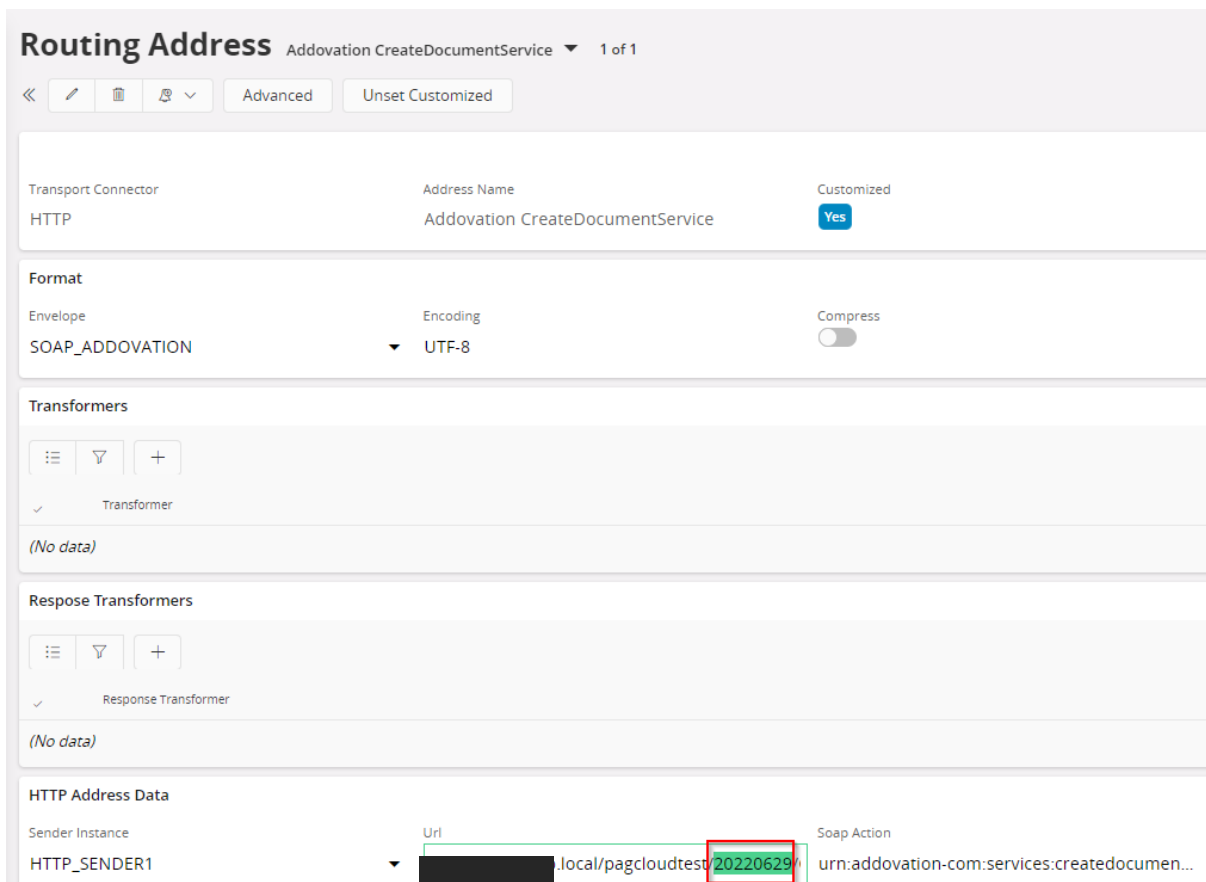


The screenshot shows the 'Routing Addresses' page in the Addovation Solution Manager. The left sidebar contains a menu with 'Routing Addresses' highlighted. The main area displays a table of routing addresses. The 'Addovation CreateDocumentService' entry is selected, and its 'Details' button is highlighted with a red box.

Address Name
AC_REST_ADDRESS
AC_REST_ADDRESS_CO-DEPLOYED
AC_REST_ADDRESS_CONFIG_SLOT
AC_REST_ADDRESS_CONFIG_SLOT_CO-DEPLOYED
AC_REST_CALLBACK
Addovation CreateDocumentService
Austrian EU Tax Sales XML

Change to new version and save:

Download zip files for on-premise installation.



The screenshot shows the 'Routing Address' configuration page for 'Addovation CreateDocumentService'. The 'URL' field is highlighted with a red box, showing the path to the service.

Routing Address Addovation CreateDocumentService 1 of 1

Transport Connector: HTTP

Address Name: Addovation CreateDocumentService

Customized: Yes

Format

Envelope: SOAP_ADDOVATION

Encoding: UTF-8

Compress: ☐

Transformers

(No data)

Response Transformers

(No data)

HTTP Address Data

Sender Instance: HTTP_SENDER1

Url: **20220629**

Soap Action: urn:addovation-com:services:createdocumen...

AIS CreateDocumentService has now been updated.