

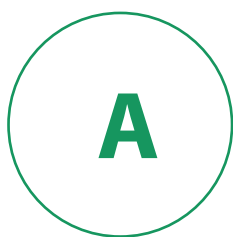


SDL Content Manager upgrade guide

SDL Content Manager 14.0.0

July 2019





Legal notice

Copyright and trademark information relating to this product release.

Copyright © 2003–2019 SDL Group.

SDL Group means SDL PLC. and its subsidiaries and affiliates. All intellectual property rights contained herein are the sole and exclusive rights of SDL Group. All references to SDL or SDL Group shall mean SDL PLC. and its subsidiaries and affiliates details of which can be obtained upon written request.

All rights reserved. Unless explicitly stated otherwise, all intellectual property rights including those in copyright in the content of this website and documentation are owned by or controlled for these purposes by SDL Group. Except as otherwise expressly permitted hereunder or in accordance with copyright legislation, the content of this site, and/or the documentation may not be copied, reproduced, republished, downloaded, posted, broadcast or transmitted in any way without the express written permission of SDL.

SDL Tridion Docs is a registered trademark of SDL Group. All other trademarks are the property of their respective owners. The names of other companies and products mentioned herein may be the trademarks of their respective owners. Unless stated to the contrary, no association with any other company or product is intended or should be inferred.

This product may include open source or similar third-party software, details of which can be found by clicking the following link: "Acknowledgments" on page 91.

Although SDL Group takes all reasonable measures to provide accurate and comprehensive information about the product, this information is provided as-is and all warranties, conditions or other terms concerning the documentation whether express or implied by statute, common law or otherwise (including those relating to satisfactory quality and fitness for purposes) are excluded to the extent permitted by law.

To the maximum extent permitted by law, SDL Group shall not be liable in contract, tort (including negligence or breach of statutory duty) or otherwise for any loss, injury, claim liability or damage of any kind or arising out of, or in connection with, the use or performance of the Software Documentation even if such losses and/or damages were foreseen, foreseeable or known, for: (a) loss of, damage to or corruption of data, (b) economic loss, (c) loss of actual or anticipated profits, (d) loss of business revenue, (e) loss of anticipated savings, (f) loss of business, (g) loss of opportunity, (h) loss of goodwill, or (i) any indirect, special, incidental or consequential loss or damage howsoever caused.

All Third Party Software is licensed "as is." Licensor makes no warranties, express, implied, statutory or otherwise with respect to the Third Party Software, and expressly disclaims all implied warranties of non-infringement, merchantability and fitness for a particular purpose. **In no event will Licensor be liable for any damages, including loss of data, lost profits, cost of cover or other special, incidental, consequential, direct, actual, general or indirect damages arising from the use of the Third Party Software or accompanying materials, however caused and on any theory of liability. This limitation will apply even if Licensor has been advised of the possibility of such damage. The parties acknowledge that this is a reasonable allocation of risk.**

Information in this documentation, including any URL and other Internet website references, is subject to change without notice. Without limiting the rights under copyright, no part of this may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of SDL Group.

1	Legal notice	3
2	Welcome to Content Manager Upgrade Guide	1
	Customer support	2
3	Upgrading Content Manager	3
	Upgrade preparation	4
	Backing up the project	4
	Backup the database	5
	Backing up Microsoft SQL Server	5
	Backing up Oracle RDBMs	6
	Backing up the closed data files	6
	Exporting for an Oracle database	7
	Uninstalling Content Manager	8
	Installing the Content Manager server	8
	Configuring database connection strings	9
	Creating and testing Oracle RDBMs	9
	Editing the tnsnames.ora database connection file	9
	Creating and testing the connection for Oracle RDBMs	10
	Creating and testing the connection for Microsoft SQL Server	11
	inputparameters.xml	12
	Mandatory input parameters	13
	Optional input parameters	14
	Executing the InstallTool	16
	Identifying reverse proxies	16
	Post upgrade tasks	17
	Verifying Microsoft SQL System Administration role permissions	17
	Running DBUpgradeTool for maintenance	18
	Troubleshooting DBUpgradeTool	19
	Completing the installation or upgrade with ISHDeploy	19
	Adding the relying party entries for webUI and WCF Services for commercial STS	20
	Removing the relying party entries for webUI and WCF Services	21
	Configure Security Token Service	21
	Content Manager Security Token Service requirements	22
	ISHSTS with Windows Authentication	24
	Configure application server for Windows Authentication	25
	Configure SQL Server database for Windows Authentication	25

The Administrator setup	26
Completing the Administrator setup	27
Validation XML configuration files	28
Upgrading Content Manager from 13.0.0 or 13.0.1	28
Upgrading the templates	28
Upgrading templates for publications in combined languages	29
Cleaning up the EDTs	30
Publishing with Content Delivery	30
Configuring Content Manager for the new translation review flow	31
Using ISHDeploy for Content Manager configuration	32
Checking the TRANSLATORSERVICE role	32
Translation management integration configuration	33
Translation services configuration files	33
Application Configuration for TranslationBuilder	33
Application configuration for TranslationOrganizer	34
TranslationOrganizer common parameters	34
TranslationOrganizer WorldServer parameters	36
TranslationOrganizer TMS parameters	37
TranslationOrganizer File System settings	40
Translation Job Workflow	41
Integration requirements for Content Manager and SDL WorldServer	41
Integration requirements for Content Manager and SDL TMS	43
Integration requirements for Content Manager and File System	45
Enabling services	46
Enabling Draft Space	46
Rebuilding the full text index	47
Configuring Content Manager for Collaborative Review	47
Specify the Collaborative Review URI in Content Manager	48
Modify the SDL Tridion Docs output type settings	48
Verifying the installation	49
Verifying URLs	50
Verify read access to the database by viewing an inbox	50
Verify read and write access to the database by creating a folder	51
Creating an account and connecting to the Repository	51
Running a client tool	52
Testing publishing	52
Executing the full text search	53
Verifying customer specific components	53
Collaborative Review upgrade	54
System management reference for Content Manager upgrade	54

Introduction to SDL Tridion Docs scalability	54
SDL Tridion Docs network cluster	54
SDL Tridion Docs environment with ISHSTS	55
SDL Tridion Docs environment with ADFS	57
Content Manager scalability	60
One server for all roles	60
Simple cluster	60
Multi server deployment	64
Advanced server cluster	65
Server roles	71
Web role	72
Full text indexing role	72
Default background task role	73
Translation role	74
Publish role	75
Best practices for creating a Trisoft InfoShare BackgroundTask service with a specific role	76
How to create a new BackgroundTask service with a role	77
How to adapt the role of an existing BackgroundTask service	78
Firewalls and blocked ports	79
SMTP	79
Microsoft SQL Server ports	80
Oracle RDBMS	81
HTTPS (SSL)	82
Background task component	82
Overview of the background task configuration	83
Usage of variables inside the background task configuration	84
Understanding the availability matrix	86
Understanding the isolation level of the handler	87
Installing desktop client tools	88
Installing the Authoring Bridge	88
Installing Publication Manager	89
Installing Condition Manager	89
Installing Content Importer	90
A Acknowledgments	91



1

Welcome to Content Manager Upgrade Guide

This document presents a description of all needed checks and actions necessary for an upgrade from the previous version of Content Manager to the latest version.

Customer support

To contact Technical Support, connect to the Customer Support Web Portal at <https://gateway.sdl.com> and log a case for your SDL product. You need an account to log a case. If you do not have an account, contact your company's SDL Support Account Administrator.

2

Upgrading Content Manager

These steps apply to Content Manager upgrade specifically.

The details provided in this chapter apply to an upgrade of the Content Manager application only, in a situation when the database system and the operating system are not upgraded.

The earliest version from which you can upgrade to the current Content Manager version is Content Manager 10.0.0. To upgrade from an earlier version, you must first upgrade to Content Manager 12.0.0 (and perform the post-upgrade operations).

Upgrade preparation

The pre-upgrade tasks are essentially a backup and uninstallation of the old Content Manager version. You should always make a complete backup of both project and database before upgrading so you can restore to the current environment if necessary.

Note:

- Before beginning the upgrade it is recommended that you warn all users that the system will be temporarily unavailable.
 - No updates should be allowed beyond this point in time.
 - You should plan to copy the whole of the Content Manager environment to a secure destination.
 - The whole of the Content Manager environment includes all Content Manager websites, Content Manager website components, the Full Text Collection, the database backup(s), and the Content Manager registry keys.
 - Identify and plan to backup any custom files and information in the same way.
 - Remember all Trisoft COM+ Applications and all Virtual Directories will have to be removed as well.
-

Backing up the project

This backup operation makes a copy of everything installed and generated in this project.

About this task

This does a backup of the business components and used configuration and makes a full copy of:

- the Full Text Collection
 - all PublishService, ImportService, ExportService generated data
 - all generated logging
-

Note: This backup can take a lot of time. You can do an initial clean up of all data that needs no backup or do a manual backup of the necessary files and settings.

Procedure

1. Login to the Windows system as the **Administrator** user or a user with Administrator privileges (a user in the Administrator group).
2. In Windows Explorer, go to
`C:\IShCD\yyyymmdd.CD.InfoShare<version_num>.ProjectName.IT__InstallTool`
3. Locate and double-click on **InstallTool.exe**.
4. Select the backup option.
5. Select the project that you want to backup and a location where you can safely store everything.

Backup the database

Follow the procedure to backup the Microsoft SQL Server or Oracle RDBMs database based on your current configuration.

Note: The complexity of the Oracle backup is much greater than that of the SQL Server. The the SQL Server algorithm is described by its Windows UI. The Oracle procedure is explained by command line tools so that it is valid for both a Windows and UNIX environment).

Backing up Microsoft SQL Server

Follow this procedure to backup your SQLServer.

Procedure

1. Open **SQL Server Management Studio**.
2. Open the **Databases** folder.
3. Right click on your database
4. Select **Tasks > Back Up**.
5. **Backup type** should be **Full**.
6. Enter a **Name** for the backup.
 It is recommended that you use the format: `yyyymmdd.projectname.bak` to name the backup.
7. Ensure that **Backup set will expire: After 0 days**
8. In the **Destination** pane at the bottom of the window, click **Add**.
9. Add a destination folder such as: `yyyymmdd.projectname\SQLServer\Dump`.
10. Click **Options** in the left pane and verify that the information is accurate:
 - Backup to the existing media set.
 - Append to the existing backup set.
 - Verify backup when finished.
11. Click **OK**.
 The backup of the database starts.

Backing up Oracle RDBMs

The description makes use of command line programs so that a person with a mixture of Windows and UNIX knowledge can follow the procedure on any Oracle hosting machine.

This procedure guides a knowledgeable person through the configuration so that no important steps are forgotten. However, it does not provide an explanations, or all the options for each step.

The procedure includes suggestions about how to do the steps. It is out of scope of this document to give a step-by-step explanation on basic Oracle maintenance. An Oracle DBA can choose any option desired, as long as an exact restoration of the Content Manager environment can be made.

Note:

- The description refers to a Windows environment concerning system variables and file paths.
 - The default database name is ISH.
 - The procedure is different depending on the Oracle version you will import back in.
-

Backing up the closed data files

You can just make a copy of closed data files. This action applies to cases when you are not planning to change the location of the files, the character set or the Oracle version, so you won't need to perform a complete export.

Procedure

1. For making a copy of closed data files, open a command line window and set the ORACLE_SID and ORACLE_HOME environment variables. For example, on Windows:

```
set ORACLE_SID=ISH
set ORACLE_HOME=C:\Oracle\Product\<version>\db_1
```

2. Start SQLPLUS /NOLOG:

```
SQLPLUS /NOLOG
```

3. Force a database to close by typing the following. The goal is to do this quickly enough so that no-one can make a connection and alter the database, ensuring that you copy a stable file version.

```
shutdown abort
startup open
shutdown normal
```

All pending connections are closed by force. The **startup open** checks the database and the **shutdown normal** closes all database files.

Exporting for an Oracle database

A full database export is the way to go when you need to be independent from version, character set or location. Exporting from the previous Oracle version for importing to the latest supported version requires a specific procedure.

Procedure

1. Copy the file `expdp.par` from `C:\ISHCD\yyyymmdd.CD.InfoShare<version_num>.\ProjectName.IT\Database\Dump\Oracle\expdp` to the dump directory (`DATA_PUMP_DIR`) on the database server.

Note: The dump directory should be in the line of `C:\Oracle\admin\<DatabaseName>\dpdump`.

2. Goto the database server.
3. Adapt the file names for the `DUMPFILE` and `LOGFILE` in the `expdp.par` file.
4. Open a command prompt.
5. Set `ORACLE_SID`: `ORACLE_SID=ISH`
6. Set `ORACLE_HOME`:
 - Oracle 12.1: `ORACLE_HOME=c:\oracle\product\12.1.0\dbhome_1`
 - Oracle 12.2: `ORACLE_HOME=c:\oracle\product\12.2.0\dbhome_1`

Note: All commands must be entered in the same window, ensuring that every environment variable set is available throughout the process. The next steps presume that you use the same shell with these variables set correctly.

7. Start `sqlplus` as `sysdba`
8. Check if there is a dump directory `DATA_PUMP_DIR`

```
SELECT directory_path FROM dba_directories WHERE directory_name = 'DATA_PUMP_DIR';
```

9. Make sure the dedicated `isource` database user has read and write access to the directory

```
GRANT read, write ON DIRECTORY data_pump_dir TO isource;
```

10. Exit `SQLPlus`, it gets you back at the command prompt
11. Execute the following command

```
expdp parfile="C:\Oracle\admin\<DatabaseName>\dpdump\expdp.par"
```

12. Provide the user name and password for the dedicated `isource` database user.
The dump file is exported in the dump directory, you can import it back after you have uninstalled the old version of Content Manager.

Uninstalling Content Manager

Removal of all Content Manager items of the installed environment is required before starting an upgrade.

Before you begin

In the context of a Content Manager install or uninstall operation, before you run InstallTool you need to make sure the Microsoft Distributed Transaction Coordinator MSDTC service is running.

About this task

All items which were installed and in the `installplan.xml` are removed from your system. Items which are not removed are:

- the Full Text Collection,
- all PublishService, ImportService, ExportService generated data,
- all generated logging,
- files that have changed after the initial installation (such as configuration files).

Procedure

1. Login to your Windows system as the **Administrator** user or a user with Administrator privileges (a user in the Administrator group).
2. In Windows Explorer, go to the location where the previous installation software is located, or to the CD that was used to install the current environment, to access the correct version of the tool for uninstalling the software. Go to:

```
C:\IshCD\yyyymmdd.CD.InfoShare<version_num>.ProjectName.IT\__InstallTool
```

3. Locate and double-click on **InstallTool.exe**.
4. Select the uninstall option.
5. Select the project that you want to remove.
6. Uninstall.

Note: If the tool ends with errors you can find extensive info in `installtool.log`.

7. After the uninstall have completed, check that no Trisoft COM+ application remained in **Component Services > Computers > My Computer > COM+ Applications**. Remove remaining applications having a name starting with Trisoft-, if any.

Installing the Content Manager server

You must install and configure the application and web server to use the Content Manager application, allowing you to connect to the repository.

Most of the Content Manager server installation process is automated. At the end of the process, some options must be set manually.

Be sure to check the installation requirements before initiating the application server installation.

Configuring database connection strings

Use the Microsoft Universal Data Link tool to create and test a connection string.

An SQL Server oriented environment makes use of **Microsoft OLE DB Driver for SQL Server** (MSOLEDBSQL).

An Oracle oriented environment needs to make use of **Oracle Provider for OLE DB**. The introduction of Unicode requires the use of this provider together with the **Enhanced Oracle Services for Microsoft Transaction Server**.

Creating and testing Oracle RDBMs

To create and test the Oracle RDBM, you must configure the `tnsnames.ora` file so that a connection can be made.

Follow the steps for editing the `tnsnames.ora` file before creating and testing the connection.

Editing the `tnsnames.ora` database connection file

The Oracle `tnsnames.ora` file must be modified on the Oracle database server and on systems that communicate with the Oracle database server. This file defines the information for a connection to the database server and to the database instance for the Content Manager repository.

About this task

If the Oracle database server and client software was installed on the same system, you must edit the `tnsnames.ora` file under each instance of `Oracle_home`.

Procedure

1. Login to the server as an administrator user.
2. Open the `Oracle_home\network\admin\tnsnames.ora` file for editing.
If the file does not exist create an empty text document named `tnsnames.ora` in the directory above.
3. Add the following to the file. Make sure that it is left-aligned (that is, no leading whitespace on the first line)

```
net_service_name =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)
        (HOST = hostname)
        (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = service_name)
    )
  )
```

where:

- `net_service_name` is an alias that is used for a connect descriptor. For example:

```
ISH.WORLD
```

- *hostname* is the IP address or name of the database server. For example:

```
(HOST = devserver01)
```

or

```
(HOST = 127.0.0.1)
```

- *SERVICE_NAME* is a combination of the *db_name* and *db_domain* values in the `C:\oracle\admin\ISH\pfile\PFIL\initISH.ora` file. For example:

```
SERVICE_NAME = ISH.ORASERVER.DOMAIN.NAME
```

where *ISH* is the *db_name* value and *ORASERVER.DOMAIN.NAME* is the *db_domain* value in the `initISH.ora` file.

4. Save and close the file.

Creating and testing the connection for Oracle RDBMs

An Oracle oriented environment makes use of Oracle Provider for OLE DB.

Before you begin

You must have a valid `tnsnames.ora` file in place to create and test the connection. If this has not been done, refer to the procedure for editing the `tnsnames.ora` database connection file.

Procedure

1. Create a new text document in your installation directory (`C:\InfoShare`) and name it `connection.udl`.
Note:
 - Be sure that the file extension is `.udl`. If required, change your Windows Explorer settings to recognize the file extension. You have created a Universal Data Link file, which has a wizard-like program associated to create connection strings.
 - Ensure that the filename was not appended with a `.txt` filename extension. If file name extensions are not visible, modify the Windows **Tools > Folder Options > View**, and check that **Hide extensions for known file types** is not selected.
2. Start the associated program by executing the command:

```
%WINDIR%\SysWOW64\cmd.exe /c START C:\InfoShare\connection.udl
```

 The command window launched from SysWOW64 ensures that the 32-bit database provider in the wizard screen is found.
3. Click the **Provider** tab, and select **Oracle Provider for OLE DB**.
4. Click **Next**.
5. Under the **Connection** tab, in the first field select or enter the data source. For example: `ISH.WORLD`.
The data source is the `net_service_name` entry in the `Oracle_home\network\admin\tnsnames.ora` file.
6. In the second field, select the **Use a specific username and password** checkbox.

- a. In the **User name** field, enter `isource`.
 - b. In the **Password** field, enter the password for the isource user.
The default password is `isource`. If you did not use the default password, be sure to enter the password used when you created the new user. The password was specified for the isource user when configuring the Oracle server.
 - c. Select the **Allow saving password** checkbox.
7. Test the connection by clicking **Test Connection**.
If the connection is valid, a message is displayed to notify that the connection is working correctly.
 8. Click **OK** to confirm.
This saves the password as plain text in the UDL file. If the connect string does not contain the password variable, no valid connections can be made with this connection string.

Creating and testing the connection for Microsoft SQL Server

An SQL Server oriented environment makes use of the **Microsoft OLE DB Driver for SQL Server**.

Procedure

1. Create a new text document in your installation directory (`C:\InfoShare`) and name it `connection.udl`.
Note:
 - Be sure that the file extension is `.udl`. If required, change your Windows Explorer settings to recognize the file extension. You have created a Universal Data Link file, which has a wizard-like program associated to create connection strings.
 - Ensure that the filename was not appended with a `.txt` file extension. If file extensions are not visible, modify the Windows **Tools > Folder Options > View**, and check that **Hide extensions for known file types** is not selected.
2. Start the associated program from Windows Explorer by double-clicking `C:\InfoShare\connection.udl`.
3. Click the **Provider** tab, and select **Microsoft OLE DB Driver for SQL Server**.
4. Click **Next**.
5. Under the **Connection** tab, in the first field select or enter the SQL database server name.
6. In the second field, select the **Use a specific username and password** checkbox.
 - a. In the **User name** field, enter `isource`.
 - b. In the **Password** field, enter the password for the isource user.
The default password is `isource`. If you did not use the default password, be sure to enter the password used when you created the new user. The password was specified for the isource user when configuring the Microsoft SQL server.
 - c. Select the **Allow saving password** checkbox.
7. In the third field, select the **Select the database on the server** checkbox, and specify the database name.
This is the name specified when you created the database. The database name was specified when configuring Microsoft SQL server.
8. Test the connection by clicking **Test Connection**.
If the connection is valid, a message is displayed to notify that the connection is working correctly.

9. Click **OK** to confirm.

This saves the password as plain text in the UDL file. If the connect string does not contain the password variable, no valid connections can be made with this connection string.

Results

When using this connection string in `inputparameter.xml`, remove the last two keywords: `Initial File Name` and `Server SPN`.

inputparameters.xml

The `inputparameters.xml` file stores key parameters that are used by the Content Manager installer. This file must accurately reflect your environment for the install tool to work correctly. Only the mandatory parameters need to be considered in the vast majority of installations. The optional parameters cover rare and advanced cases.

Overview

Note: Before modifying the `inputparameters.xml` file, you need to

- Obtain and install the certificate.
 - Configure HTTPS bindings.
-

Each parameter in this configuration file has the following syntax:

```
<param name="parameter">
  <currentvalue>value</currentvalue>
  <defaultvalue>example_value</defaultvalue>
  <description>description_of_how_used</description>
  <validate>if_validated</validate>
</param>
```

The XML elements perform the following functions:

<currentvalue>

Contains the value that is used by the Content Manager installer.

<defaultvalue>

Contains a predefined value as an example. Do not use the predefined value as a default value; its only purpose is to serve as an example.

<description>

Contains details describing how the current value of the parameter is used.

<validate>

Defines whether the value of the parameter is validated or not. If the element is empty, no validation is performed. The `<validate>` values are provided, and they should not be modified.

Mandatory input parameters

This section lists and describes the main input parameters contained in the `inputparameters.xml` file, the ones you need to set for any installation. Some of these input parameters are mandatory: you need to set them for the installation to work correctly.

osuser

The user name for the designated operating system account.

ospassword

The password for the designated operating system account. Set the password so that it never expires.

connectstring

The connection string for the instance of the database application.

Note: To obtain the required `connectstring` parameter:

- In a text editor, for example Notepad, open the `connection.udl` file you created in the installation directory (typically, `C:\InfoShare`).
- Browse to the line beginning with `Provider=`.
For Oracle:

```
Provider=OraOLEDB.Oracle.1;Password=isource-password;Persist Security
Info=True;User ID=isource;Data Source=data_source_name
```

For MSSQL:

```
Provider=SQLOLEDB.1;Password=isource-password;Persist Security
Info=True;User ID=isource;Initial Catalog=database;Data Source=server-
name
```

- Copy the line and paste it in the `inputparameters.xml` file to specify the `connectstring` parameter.
-

apppath

The root directory for the Content Manager installation.

By default, this is set to `C:\InfoShare`.

ps_java_home

The path to the `JAVA_HOME` directory.

serviceusername

Specify the user name for the Content Manager services.

servicepassword

Specify the password matching `serviceusername`'s value.

Optional input parameters

This section lists and describes the optional input parameters contained in the `inputparameters.xml` file. The value of several optional parameters are calculated by Content Manager, others have a default value. These parameters should be changed only by an expert of Content Manager.

Optional parameters

Note: When you provide a value to a parameter that is calculated by default, the parameter is set to your value and is no longer calculated.

Name	Description	Remarks
projectsuffix	A suffix that specifies the particular instance of Content Manager, if you are installing multiple Content Manager instances on a server.	By default: no value
webpath	The root directory for the Web site.	By default: <apppath value> (calculated)
datapath	The directory containing the data directory. This directory stores all data exported from Content Manager	By default: <apppath value> (calculated)
workspacepath	The temporary directory where installation files are stored before installation.	By default: <apppath value>_Workspace (calculated)
infoshareauthorwebappname	The name of the web client application.	By default: ISHCM + <projectsuffix value>
infosharewswebappname	The name of the web services application.	By default: ISHWS + <projectsuffix value>
infosharestswebappname	The name of the website with the Content Manager Secure Token Service (STS).	By default: ISHSTS + <projectsuffix value>
websitename	The name of the web site where all virtual directories are created.	By default: Default Web Site
baseurl	The base URL that is used to access the Content Manager web client.	By default: extracted from the Common Name of the certificate specified for HTTPS binding on IIS (calculated).
localservicehostname	The hostname for the local address: The host name with which communication within the Content Manager box takes place.	By default: the NetBIOS name of this local computer (calculated)

Name	Description	Remarks
ps_fo_processor	The fully qualified file name of the XSL-FO processor, for example C:\Program Files\AntennaHouse\AHFormatterV6\AHFCmd.exe.	By default: the highest installed version from %ProgramFiles% (calculated).
ps_htmlhelp_processor	The fully qualified file name of the HTML Help processor.	By default: C:\Program Files (x86)\HTML Help Workshop\hhc.exe Note: Do not use environment variables or quotation marks.
ps_java_jvmdll	The full path to jvm.dll inside the openJDK JAVA_HOME folder, used by the TrisoftSolrLucene service.	By default: <ps_java_home value>\bin\server\jvm.dll Note: do not use environment variables or quotation marks.
ps_javaahelp_home	The full path to a JavaHelp JHHOME folder, used when you want to publish with the JavaHelp output type.	By default: C:\javaahelp\jh2.0 Note: do not use environment variables or quotation marks.
ps_webworks_automap_application	The fully qualified path and filename of WebWorks .	By default: No value Note: do not use environment variables or quotation marks.
solrlucene_service_port	The port the SolrLucene service uses. This port must be unique for each Content Manager instance installed on a server.	By default: 8078
solrlucene_stop_port	The port that is used to stop the SolrLucene service. This port must be unique for each Content Manager instance installed on a server.	By default: 8077
basehostname	The hostname part of the base URL, calculated using <i>baseurl</i> as input stripping the protocol HTTP/HTTPS (e.g. TRIDT06.tribe.trisoft.be)	By default: from <i>baseurl</i> (calculated)
infosharestswindowsauthenticationenabled	Whether or not the infosharests web site enables IIS windows authentication.	By default: False

Name	Description	Remarks
servicecertificatethumbprint	The thumbprint of the service certificate (can be a re-purposed SSL certificate). All the linked web/app servers in a farm must offer the same trusted certificate.	By default: from HTTPS binding of websitename
servicecertificatevalidationmode	The validation mode of the service certificate (servicecertificatethumbprint).	By default: ChainTrust
servicecertificatesubjectname	The subject name of an already installed certificate for unique service identification (can be a re-purposed SSL certificate). All the linked web/app servers in a farm must offer the same trusted certificate.	By default: from HTTPS binding of websitename

Executing the InstallTool

Use the Content Manager InstallTool to install and merge all standard and customer specific project files.

Before you begin

In the context of a Content Manager install or uninstall operation, before you run InstallTool you need to make sure the Microsoft Distributed Transaction Coordinator MSDTC service is running.

Procedure

1. Login to your Windows system as a Content Manager user with the Administrator user role.
2. In Windows Explorer, go to
C:\ISHCD\yyyymmdd.CD.InfoShare<version_num>\ProjectName.IT__InstallTool
3. Locate and double-click on **InstallTool.exe**.
4. Select the **Install** option by entering the number **2**.
5. Hit enter to respond to all questions; the default options should be sufficient.

All standard and customer specific project files are merged and installed. All required services are up and running (the Crawler, SolrLucene, IISAdmin, W3SVC).

Identifying reverse proxies

Various reverse proxy servers can be used in combination with Content Manager. Using reverse proxy servers may result in connection problems to the Content Manager repository. You can check the BASEURL to verify that it is not causing a problem in Internet Explorer.

When you identify reverse proxy servers, verify that the following URLs do not cause any problems in Internet Explorer:

- BASEURL/ISHWS/Application.asmx?wsdl

- `BASEURL/ISHWS/Application.asmx?disco`

The `BASEURL` is specified in the `inputparameters.xml` file used by the Content Manager installer.

The `BASEURL` must use the `https` schema. It must also reference a hostname that is valid for the SSL certificate.

For example if the server has a SSL certificate configured with `CN=example.com` then the `BASEURL` must be `https://example.com`.

Example:

```
<param name="baseurl">
<currentvalue>https://example.com</currentvalue>
<defaultvalue>https://example.com</defaultvalue>
<validate/>
</param>
```

Make sure that the URLs are accessible by your users.

Post upgrade tasks

After you complete the upgrade of your Content Manager server, you should perform a number of post-upgrade tasks. These tasks help you to verify the installation and configure the components that you have just upgraded.

Verifying Microsoft SQL System Administration role permissions

To ensure that the database upgrade tool (DBUpgradeTool or DBUT) works properly, the `isource` user must have system administrator permissions.

About this task

To allow DBUT to fully execute all necessary tasks to update your MS SQL database now and for each new release, follow the procedure below to ensure the database user has the necessary permissions.

Note: This procedure applies only if you are using SQLServer. This procedure does not apply to Oracle.

Procedure

1. Access the **SQL Server Management Studio**.

Windows 2012: To access the **SQL Server Management Studio** if not readily accessible, use the **Windows Powershell** icon on the bottom toolbar then at the prompt type: **Ssms.exe**.

Note: If prompted and required, connect to the server.

2. Under the folder for the MSSQL server in the left pane, open **Security > Logins**.
3. Right-click on the **isource** user and select **Properties**.

A **Login Properties - isource** window displays.

4. In the left pane of the **Login Properties - isource** window select **Server Roles**.
5. In the right pane for Server Roles select **public** and **sysadmin**.
6. Click **OK**.
7. Click **File > Exit** to exit and close the Microsoft Server Management window.

Running DBUpgradeTool for maintenance

DBUpgradeTool (DBUT), for all supported database engines, performs an overall verification of the database and updates database objects such as views, indexes, packages and stored procedures and modifies metadata structures.

Before you begin

- System Administrator rights for the *isource* database user.
- Available, complete and correctly installed Content Manager server and Content Manager database (up and running).
The installation ensures that the upgrade files and connections are in place to allow a successful upgrade.
- Exclusive access to the Content Manager database for DBUT.
Be sure to stop all components and services, such as InfoShare Crawler, on all servers.

Procedure

1. Login to the server as a Windows user with the Administrator user role.
2. In Windows Explorer, in the Content Manager installation directory, open: `\ App\ Setup\ DBUpgradeTool\`
3. Locate and double-click on `DBUpgradeTool.exe`
4. Optional, for ADFS only:
Use DBUT to configure the FishExternalID of the administrator user with a value such as *domain\username*, for the user who has to log in the system as administrator.
 - Run DBUpgradeTool, select option 1: **Maintenance**.
 - Select option 5: **Append** an user's externalid for initial STS login.
 - Choose the installation and then fill in the appropriate value for the Content Manager admin user.
5. Select the **DatabaseUpgrade** option.
6. Hit the **Enter** key to respond to all questions; the default options are sufficient for the following questions:
 - Select your recently installed project (thereby selecting the database location and user, upgrade script paths, version and so on).
 - Default for the application.The result is an upgrade from an older or same version to the installed version when applicable. The application pool needs to be restarted after a DBUT run.

Troubleshooting DBUpgradeTool

You may encounter issues during DBUT execution for database upgrade purposes.

Execution is interrupted

If the DBUT suddenly dies, it results in a hanging database upgrade logged in to the database. All other attempts to upgrade will fail since only one upgrade process is permitted to run at a time.

To force an undo of a hanging update, execute DBUT then select **Maintenance > Terminate** and your current project.

Current user isource warnings

The following warnings can occur in DBUT:

- WARNING: The current user "isource" does not have permission to disable the standard database job for InfoShare.
- WARNING: The current user "isource" does not have permission to install the standard database job for InfoShare. Ask a System Administrator to manually execute ISH_CreateStandardInfoShareJob.sql and ISH_EnableStandardInfoShareJob.Sql to create and enable the standard InfoShare job!
- WARNING: The current user "isource" does not have permission to enable the standard database job for InfoShare.

These warnings are returned when the user executing DBUT has not enough rights to access/create database jobs.

- Creating a database job requires a database administrator to execute `CD-Package\ Database\ Common\<SQLServerVersion>\Create\ISH_CreateStandardInfoShareJob.sql`
- Enabling a database job requires a database administrator to execute `CD-Package\ Database\ Common\<SQLServerVersion>\Create\ISH_EnableStandardInfoShareJob.Sql`

Requesting assistance

In case you need to request assistance, you need to provide the following information:

- A screenshot of the failed execution of DBUpgradeTool.
- The log file: `\App\Setup\DBUpgradeTool\DBUpgradeTool1.log`

Completing the installation or upgrade with ISHDeploy

Most post-install/upgrade tasks can be carried out with ISHDeploy.

Always configure Windows Authentication with ISHDeploy.

You can also replace most of the manual configuration tasks described in the installation and upgrade sections of this documentation by the use of commandlets.

The presentation and documentation for ISHDeploy can be found [here](#).

Adding the relying party entries for webUI and WCF Services for commercial STS

After the Content Manager installation you have to create the Relying Party Trust for the website and the WCF services on the STS Server if you are not using ISHSTS. This is required to allow logins to Content Manager.

About this task

These post-installation steps have to be done when you are using a commercial STS such as ADFS (ADFS is used as example here). If you are using ISHSTS as STS, do not follow these steps.

Doing this manually can be error prone, therefore SDL provides you a PowerShell script that creates the Relying Party Trust and does the configuration. To execute the script do the following:

Procedure

1. The ADFS server needs the service certificate that is used by the Content Manager WCF Services. This is the same value as the certificate whose thumbprint is used in the inputparameters parameter `servicecertificatethumbprint`. With the typical Content Manager setup, this WCF Service certificate is the same as the IIS Website Certificate for SSL.
 - a. Open **Internet Information Services (IIS) Manager***Servername*.
 - b. Double-click **Server Certificates** in the right pane.
 - c. Right-click on the certificate of the IIS website that is going to be used for Content Manager then click **View**.
 - d. Click on the **Details** tab then click **Copy to File** and export the certificate to the file system (only export the public key) e.g. *SDL.ISH.cer*.
2. Copy the PowerShell scripts which are created in the directory `\InfoShare\App\Setup\STS\ADFS\Scripts\scripts` to a temporary directory on the ADFS server e.g. `C:\SDL.ISH`. Copy also the certificate `C:\SDL.ISH` from the previous step.
3. Login into the ADFS Server and open a administrative PowerShell command line; right-click on the PowerShell shortcut and choose **Run as administrator**.
4. Set the PowerShell execution policy to `Unrestricted`.

Note: The scripts provided are not signed because they are generated during the Content Manager installation. To execute unsigned scripts in PowerShell you must set the execution policy to `Unrestricted`.

- To check if PowerShell's execution policy is already set to `unrestricted`, execute the command:

```
Get-ExecutionPolicy
```

- If the output of this command shows something other than `Unrestricted`, execute the command:

```
Set-ExecutionPolicy Unrestricted
```

5. Change the working directory of the command prompt by executing the command:

```
cd c:\SDL.ISH
```

- Before running the script, load the ADFS PowerShell module by issuing the command:

```
Import-Module ADFS
```

- Execute the command:

```
.\SDL.ISH-ADFSv3.0-RP-Install.ps1 "C:\SDL.ISH\SDL.ISH.cer"
```

Now if you open **AD FS Management** you should see two new Relying Party entries with the base URLs you use for the Content Manager instance.

Removing the relying party entries for webUI and WCF Services

If you want to remove the Relying Party entries, for instance because Content Manager is moved to another server or the URL has changed, follow the steps below.

Procedure

- Check if the Uninstall script for the Relying Party entries is on the ADFS server. If not, copy it from the Content Manager application server directory `\InfoShare\App\Setup\STS\ADFS\Scripts\`.
- Login into the ADFS Server and open a administrative PowerShell command line; right-click on the PowerShell shortcut and choose **Run as administrator**.
- Before running the script, load the ADFS PowerShell module by issuing the command:

```
Import-Module ADFS
```

- Execute the command:

```
.\SDL.ISH-ADFSv3.0-RP-UnInstall.ps1
```

Now if you open **AD FS Management** you should see that the two Relying Party entries are removed.

Configure Security Token Service

Client installations can be used only if you properly configure a Security Token Service (STS). The default system that manages user identity for Tridion Docs is ISHSTS.

Using ISHDeploy is an alternative to the manual tasks described in this chapter.

Content Manager Security Token Service requirements

Required identifiers and certificates for a Security Token Service configuration applicable to Content Manager.

Profiles

Content Manager relies on both the *Passive profile* and the *Active profile* to do *Federated Authentication*.

Content Manager service name	Profile	Remarks
ISHCM	Passive profile	Refers to web applications. Token encryption is optional.
ISHWS	Active profile	Refers to SOAP-based web services implementing the WS Trust protocol. Token encryption is mandatory.

Identifiers and encryption certificates

For each service Content Manager expects the following combination of identifiers and encryption certificate to be configured on a Security Token Service.

Service: ISHCM

No encryption certificate.

Identifier: `https://example.com/ISHCM/`

Service: ISHWS

The encryption certificate is the public key of the certificate referenced through the `servicecertificateThumbprint` input parameter

Identifiers:

- `https://example.com/ISHWS/`
- `https://example.com/ISHWS/Wcf/API25/Application.svc`
- `https://example.com/ISHWS/Wcf/API25/Baseline.svc`
- `https://example.com/ISHWS/Wcf/API25/DocumentObj.svc`
- `https://example.com/ISHWS/Wcf/API25/EDT.svc`
- `https://example.com/ISHWS/Wcf/API25/EventMonitor.svc`
- `https://example.com/ISHWS/Wcf/API25/Folder.svc`
- `https://example.com/ISHWS/Wcf/API25/ListOfValues.svc`
- `https://example.com/ISHWS/Wcf/API25/MetadataBinding.svc`
- `https://example.com/ISHWS/Wcf/API25/OutputFormat.svc`
- `https://example.com/ISHWS/Wcf/API25/PublicationOutput.svc`
- `https://example.com/ISHWS/Wcf/API25/Search.svc`
- `https://example.com/ISHWS/Wcf/API25/Settings.svc`

- <https://example.com/ISHWS/Wcf/API25/TranslationJob.svc>
- <https://example.com/ISHWS/Wcf/API25/TranslationTemplate.svc>
- <https://example.com/ISHWS/Wcf/API25/User.svc>
- <https://example.com/ISHWS/Wcf/API25/UserGroup.svc>
- <https://example.com/ISHWS/Wcf/API25/UserRole.svc>
- <https://example.com/ISHWS/Wcf/API20/Application.svc>
- <https://example.com/ISHWS/Wcf/API20/DocumentObj.svc>
- <https://example.com/ISHWS/Wcf/API20/EDT.svc>
- <https://example.com/ISHWS/Wcf/API20/EventMonitor.svc>
- <https://example.com/ISHWS/Wcf/API20/Folder.svc>
- <https://example.com/ISHWS/Wcf/API20/MetaDataAssist.svc>
- <https://example.com/ISHWS/Wcf/API20/OutputFormat.svc>
- <https://example.com/ISHWS/Wcf/API20/Publication.svc>
- <https://example.com/ISHWS/Wcf/API20/PublicationOutput.svc>
- <https://example.com/ISHWS/Wcf/API20/Reports.svc>
- <https://example.com/ISHWS/Wcf/API20/Search.svc>
- <https://example.com/ISHWS/Wcf/API20/Settings.svc>
- <https://example.com/ISHWS/Wcf/API20/Workflow.svc>
- <https://example.com/ISHWS/Wcf/API/Application.svc>
- <https://example.com/ISHWS/Wcf/API/ConditionManagement.svc>

Claims in the token

Content Manager maps an incoming token to a user in the users repository by the external identifier.

The mapping is done through the token's attribute matching the claim type
<http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name>.

For a token to be useful for Content Manager, the token's subject name should match a user in the Content Manager users repository.

Token claims example

For a user that can be identified as `user@company.com`, the External ID in the users repository is expected to be `user@company.com`.

A valid incoming token must have at least the following attributes defined in it:

```
<saml:AttributeStatement>
  <saml:Attribute AttributeName="name" AttributeNamespace="http://schemas.xmlsoap.org/ws/2005/05/identity/claims">
    <saml:AttributeValue>user@company.com</saml:AttributeValue>
  </saml:Attribute>
</saml:AttributeStatement>
```

ISHSTS with Windows Authentication

You need to perform several settings before ISHSTS can provide Windows Authentication. Both server and SQL server database must be properly configured. You can either make these settings manually or use the scripts provided with the package.

ISHSTS is automatically configured through the installation.

InstallTool creates an application pool such as `TrisoftAppPoolISHSTS` based on the input parameter `infosharestswebappname`. The application pool is assigned an identity based on the input parameter `osuser`. This user is responsible for hosting the endpoints provided by ISHSTS

For Windows Authentication endpoints to work, the following changes based on the requirements of Service Principal Names defined in the Active Directory must be made, either manually or through a script.

Note: The following needs to be applied per installation server and are valid only for deployments that do not include network balanced front end servers

Application pool identity

A change of the application pool identity in order to use the integrated `ApplicationPoolIdentity`. This changes the user who hosts the endpoints to an account that the correct Service Principal Names is assigned to. The expected Service Principal Names are

- `http/baseurl`
- `host/baseurl`

Note: The new user is identified locally as `IIS AppPool\infosharestswebappname` and it requires certain permissions to access resources. When this user accesses network resources it is identified as the computer account `Domain\ Computer$` where the `Domain` and `Computer` are netbios based. e.g. `TESTDOMAIN\SERVER01$`

Read permissions

Read permissions to the token signing certificate's private key are assigned to the `IIS AppPool\infosharestswebappname`. The token signing certificate in ISHSTS is configured through the `InstallTool` parameter `issuercertificatethumbprint`

Read/write permissions to the three target installation paths defined in the input parameters are assigned to the `IIS AppPool\infosharestswebappname`:

- `webpath`
- `datapath`
- `apppath`

Integrated authentication

If the database is SQL Server and the connection string utilizes integrated authentication then we grant the computer account permissions to the database.

The only permission required is `SELECT`

Configure application server for Windows Authentication

Here is how you execute the script that configures the server for ISHSTS with Windows Authentication.

Before you begin

This task requires a PowerShell session that with `Execution Policy` set to `Unrestricted`.

If it is not set, you need to set it permanently by executing the following:

```
Set-ExecutionPolicy Unrestricted
```

The task requires administrator privileges.

Note: InstallTool has already transformed the script based on the input parameters.

Procedure

1. Locate the PowerShell script `SDL.ISH-ISHSTS-Configure for Windows Authentication.ps1` in the folder `\InfoShare\App\Setup\STS\ISHSTS\Scripts`
2. Open PowerShell with elevated administrator privileges. Run `As Administrator`.
If the PowerShell session is not running with administrator privileges, the script will launch a new session and administrator privileges will be requested to the user.
3. Navigate to the script folder `\InfoShare\App\Setup\STS\ISHSTS\Scripts`
 - a. `cd \InfoShare\App\Setup\STS\ISHSTS\Scripts`
4. Execute script `SDL.ISH-ISHSTS-Configure for Windows Authentication.ps1`
 - a. `.\SDL.ISH-ISHSTS-Configure for Windows Authentication.ps1`

PowerShell session

```
cd \InfoShare\App\Setup\STS\ISHSTS\Scripts  
& '.\SDL.ISH-ISHSTS-Configure for Windows Authentication.ps1'
```

Configure SQL Server database for Windows Authentication

Here is how you execute the script that allows the server's computer account to access a SQL Server database.

Before you begin

The task applies for SQL Server database when the connection string used integrated authentication.

The task requires sysadmin rights on the SQL Server.

Note: InstallTool has already transformed the script based on the input parameters.

Procedure

1. Locate the script `GrantComputerAccountPermissions.sql` in the `\InfoShare\App\Database\Common\` folder. Depending on your version of SQL Serve:
 - For SQL Server 2016 the script path is `\InfoShare\App<Projectsuffix>\Database\Common\SQLServer2016\Tools\GrantComputerAccountPermissions.sql`
 - For SQL Server 2017 the script path is `\InfoShare\App<Projectsuffix>\Database\Common\SQLServer2017\Tools\GrantComputerAccountPermissions.sql`
2. Execute the script on the target SQL server instance

SQL Server script that grants necessary permissions

If the target database is INFOSHAREDDB and the computer account is TESTDOMAIN\SERVER01\$ then the script looks like this:

```
USE [master]
GO
CREATE LOGIN [TESTDOMAIN\SERVER01$] FROM WINDOWS WITH DEFAULT_
DATABASE= [INFOSHAREDDB]
GO
USE [INFOSHAREDDB]
GO
CREATE USER [GLOBAL\MEDEVASARAFIA01$] FOR LOGIN [TESTDOMAIN\SERVER01$]
GO
USE [INFOSHAREDDB]
GO
GRANT SELECT TO [TESTDOMAIN\SERVER01$]
GO
```

The Administrator setup

Needs to be done only if you did not receive a fully prepared database dump-backup file, otherwise this is done and configured.

Note:

- This is the responsibility of a functional administrator not of a technical administrator.
 - The configurations handled in this section are all managed through the Author website Settings tab.
 - The delivered configuration files are available in `web\Author\EnterViaUI` or, for your customer specific files, in `\CustomerSpecificFiles\Websites\Author\EnterViaUI`.
-

Completing the Administrator setup

Needs to be done only if you did not receive a fully prepared database dump-backup file, otherwise this is done and configured.

About this task

Procedure

1. Login to the Content Manager web client as an administrator user.
2. In the web Content Manager, select the **Settings** tab.
3. To configure each of the following:
 - a. Under the **Settings** tab, go to the place as noted in the **To configure, go to** column in the table below.
 - b. Delete the contents of the textbox.
 - c. Copy the contents of the file indicated in the **Copy from** column to the textbox. When you copy, ensure that there are no leading empty lines.

The files are located on the server in the Content Manager_home directory in \ Web \ Author \ EnterViaUI

To configure, go to:	Copy the contents from:	On the top menu bar, click:
XML Inbox Settings	Admin.XMLInboxConfiguration.xml	Save
XML Write Plug-In Settings	Admin.XMLWriteObjPluginConfiguration.xml	Save
XML Publish Plug-In Settings	Admin.XMLPublishPluginConfiguration.xml	Save
XML Status Settings	Admin.XMLStatusConfiguration.xml	Save
XML Translation Settings	Admin.XMLTranslationconfiguration.xml	Save
XML ChangeTracker Settings	Admin.XMLChangeTrackerConfiguration.xml	Save
XML Background Task Settings	Admin.XMLBackgroundTaskConfiguration.xml	Save
XML Extension Settings	Admin.XMLElementExtensionConfiguration.xml	Save
XML Collective Spaces Settings	Admin.XMLCollectiveSpacesConfiguration.xml	Save

Validation XML configuration files

As part of the introduction of the Settings API 2.5, a schema was made for all configuration XML files.

- Configuration files received a version number

```
<InfoShareStates version="1.0">
...
</InfoShareStates>
```

- Configuration files are validated against this schema when they are submitted through the user interface (UI). The rest of the application assumes that the configuration files in the database are valid.

Resubmit the legacy configuration XML files using the Web Client, **Settings** tab. The configuration file is validated and some corrections are made. For example, a @version attribute with the value, 1.0 is added. If there are validation errors when resubmitting the configuration files, remove all statuses in the status definitions with value "Not found as LOV Value".

```
<Status Elm="..." value="Not found as LOV Value"/>
```

Upgrading Content Manager from 13.0.0 or 13.0.1

New features and feature changes that appeared between the 13.0.0 release and the service packs require specific actions for the upgrade.

Upgrading the templates

Upgrading the templates will provide one or several of the following results, depending on which Content Manager version you are upgrading from: replacement of Oasis DTDs by SDL DTDs, creation of new templates, addition of template languages, addition of template descriptions (in English).

About this task

This new DOCTYPE incorporates `ishcondition` attributes as well as adaptations for DITA 1.3.

This task applies to restored databases from Content Manager 12.0.x., or Content Manager 13.0.x earlier than 13.0.2.

Procedure

1. Find and open the `App\Database\Common\DatabaseIndependent\Examples\UpdateEditorTemplates` directory in Content Manager home directory.
2. Locate and run the `Update-XmlEditorTemplates.ps1` script.

Results

Material is uploaded from the `\Web\Author\EnterViaUI\System\Editor template` directory, and is used to create the following items:

- A new version for the existing templates (including the API reference specific topic type), with SDL DTDs replacing the Oasis DTDs. This also affects objects used as templates.
- New templates according to the types introduced by DITA 1.3 (such as the troubleshooting type).
- Languages for the templates: English (en), German (de), Dutch (nl), Japanese (ja) and Chinese (zh).
- A description for each default template, to be used when an English language object is created.

Upgrading templates for publications in combined languages

A publication with combined languages needs extra maps, extra indexes and specific headers. This requires that you enable specific templates.

About this task

This task applies to restored databases from Content Manager 12.0.x.

Procedure

1. Find and open the `App\Database\Common\DatabaseIndependent\Examples\CreateModuleCombineLanguages` directory in Content Manager home directory.
2. Locate and run the `CreateModuleCombineLanguages.ps1` script.

The script will create a header topic `ISHPUBLMODULECOMBINELANGUAGES` in all languages that are available in the Content Manager database using the files located in `\Web\Author\EnterViaUI\System\Publishing\Topics`.

For the Content Manager default two-letter language codes, the header topic is created using the matching file from `\Web\Author\EnterViaUI\System\Publishing\Topics`. For others language codes, the code is used as title. For example, from a `ISHPUBLMODULECOMBINELANGUAGES=1=en-us.xml`:

```
<!DOCTYPE topic PUBLIC "-//OASIS//DTD DITA Topic//EN" "topic.dtd" []>
<topic id="ISHPUBLMODULECOMBINELANGUAGES" xml:lang="en-us">
  <title>en-us</title>
  <body>
    <section></section>
  </body>
</topic>
```

During publishing the file `ISHPUBLMODULECOMBINELANGUAGES=1=en.xml` will create a section in the multi-language publication that will be labeled **English** (because `en` is a default code). On the other hand, `ISHPUBLMODULECOMBINELANGUAGES=1=en-us.xml` will create a section in the multi-language publication that will be labeled **en-us**.

3. If you want to create a user-readable label for any non default language or language code, you can either:
 - Create the corresponding XML file and specify the label inside the `<title>` element before running the script.
 - Or edit the header topic created by the script after it has been run.

Cleaning up the EDTs

The Electronic Document Types (EDT) list needs to be cleaned up with the help of SDL Support.

Some EDT has been made obsolete in the new version, and mime types have been upgraded. Contact SDL Support to get help in removing the obsolete EDTs from your system's list as well as correcting the mime types.

Publishing with Content Delivery

Publishing to Tridion Docs new delivery platform requires specific settings.

Upgrade to 13.0.x from 12.0.x

When you are planning to use Content Delivery as delivery platform with Content Manager 13.0.x, the recommended procedure is:

1. Upgrade to Content Manager 13.0.x.
2. Install Content Delivery.
3. Configure Content Manager for Content Delivery, as described in Tridion Docs documentation (specify Discovery URL, Client ID and Client Secret).

The upgrade installs new settings that will enable the Write plugin to initiate the appropriate publish process, depending on the output format you are publishing to.

When Content Manager and Content Delivery are installed and configured, before publishing your content you need to:

- Create the Dynamic Delivery output format, preferably by running the `Create-DITADLVR.ps1` script that we provide.
- Provide values for the **Product family name** and **Product release name** lists, in Content Manager **Settings**.
- For each publication you want to publish, set the appropriate **Product family name** and **Product release name**, and create its Dynamic Delivery output format.

Upgrade to 13.0.1 from 13.0.0

The output format name has changed between those two versions, therefore a label must be updated. It can be done either manually or by using a script.

- By script: run `Create-DITADLVR.ps1`. It will overwrite the output format old name with the new one.
- Manually: in **Settings > Output Formats**, open the **Properties** of the `DITA Delivery` output format and replace the `DITA Delivery` label with `Dynamic Delivery`.

Configuring Content Manager for the new translation review flow

The new translation flow requires a list of object statuses that is slightly different than the one used by the previous version's flow. During an installation, the new status list is configured out-of-the-box. During an upgrade, however, the existing list of statuses is left unchanged and you need to add the new ones manually.

The new translation review flow includes an approval/rejection step. Statuses with specific names, and transitions, need to exist for this flow to run without error with WorldServer or TMS. After a Content Manager upgrade, you need to create them prior to run the out-of-the-box new flow with WorldServer and TMS.

New statuses:

- **Translation in review**
- **Translation rejected**

Status renamed:

- **Translated** has become **Translation approved**

The list of status - role - transition - inbox combinations that need to be added to the ones that already exist in Content Manager 12.0.0. You can find comprehensive descriptions in the Workflow chapter inside the usage section of Tridion Docs documentation.

User role	Initial status: Object type	Initial status: Object status	Transition(s)	Inbox(es)
Author	All types	Draft	Draft to To be reviewed	Author receiving each map/topic/image/library with Draft, In translation or Released status.
Reviewer	All types	NONE	Translation approved to [To be translated; Released] Translation in review to [Translation approved; Translation rejected]	Reviewer receiving each map/topic/image/library with To be reviewed status. Translated receiving each object with Translation approved status.
Translator	All types	NONE	In translation to [Translation approved; To be translated] Translation approved to To be translated	In Translation receiving each object with In translation status. Translator receiving each map/topic/image/libraries with To be translated status.

User role	Initial status: Object type	Initial status: Object status	Transition(s)	Inbox(es)
TRANSLATORSERVICE	All types	To be translated	In translation to [To be translated; Translation approved; Translation in review] Translation in review to [To be translated; Translation approved; Translation rejected] Translation rejected to [To be translated; In translation]	NONE

You can also customize Content Manager workflow and use entirely different statuses and transitions than the ones provided out-of-the-box. Before you add or change statuses and transitions, make sure you add a `<translationstatuses>` element to the **XML Translation Settings**, according to the instructions provided in Tridion Docs documentation.

Using ISHDeploy for Content Manager configuration

ISHDeploy provides comandlets that let you configure Content Manager using intentions instead of procedures.

ISHDeploy provides comandlets for all basic setup operations, and many advanced configuration tasks. Although manual configuration is still available, we recommend you use ISHDeploy for improved simplicity and safety. ISHDeploy also gives you access to automation.

The presentation and documentation for ISHDeploy can be found [here](#).

Checking the TRANSLATORSERVICE role

An active TRANSLATORSERVICE role is necessary for translation services (TranslationOrganizer, TranslationBuilder) to operate. This role is part of the out-of-the-box settings for Content Manager 10.0.0 and later. Older versions require that you create it manually along with a list of status transitions.

Procedure

1. In the web client, click the **Settings** tab.
2. Click **User Roles** in the left pane.
 - If *TRANSLATORSERVICE* is part of the list, select it and click the **Properties** button. Make sure that the **Active** checkbox is ticked.
 - If *TRANSLATORSERVICE* is not part of the list, create it by clicking the **New** button and specifying *TRANSLATORSERVICE* in the **Name** field.
3. Click **Status Transitions** in the left pane.

4. Check that the following status transitions are part of the list:

From Status	To Status	User Role
To be translated	In translation	TRANSLATORSERVICE
In translation	Translation in review	TRANSLATORSERVICE
In translation	Translation approved	TRANSLATORSERVICE
Translation in review	Translation approved	TRANSLATORSERVICE
Translation in review	Translation rejected	TRANSLATORSERVICE
Translation rejected	In translation	TRANSLATORSERVICE

- If all these transitions are part of the list, no other action is required.
- If any of these transitions are not part of the list, create them by clicking the **Add** button and filling in the fields with the information from the table above. You can skip the **Condition** fields.

Translation management integration configuration

The configuration of the Translation management integration with WorldServer, TMS and File System is described.

Translation services configuration files

The configuration files for translation management are described. You can modify the files to configure or customize translation management for your needs.

Application Configuration for TranslationBuilder

Modify the `translationbuilder.exe.config` file parameters noted below to configure or customize TranslationBuilder.

The file is located on the Content Manager server: `\InfoShare\App\TranslationBuilder\Bin\translationbuilder.exe.config`

All the parameters are configured in the `<settings>` element in the `trisoft.infoShare.translationBuilder` section.

Name	Description
<code>maxObjectsInOnePushTranslation</code>	Maximum number of objects in a single push translation. The default value is 1000.
<code>maxTranslationJobItemsCreatedInOneCall</code>	Maximum number of items created in a single transaction. The default value is 10000
<code>maxTranslationJobItemsUpdatedInOneCall</code>	Maximum number of items updated in a single transaction. Default value is 100. Maximum allowed value is 999.

Name	Description
completedJobLifeSpan	The time after which the completed/cancelled job is deleted. Default value is 90.00:00:00.000
jobProcessingTimeout	The time that a job can be processed by a single step without updating the job lease before it is considered dead. Default value is 01:00:00.000.
userName	The name of the user to access Content Manager. This value is initialized from install parameters.
password	The password of the user to access Content Manager. This value is initialized from install parameters.
jobPollingInterval	Interval at which jobs are polled for processing. Default value is 00:05:00.000.
pendingJobPollingInterval	Interval at which jobs that are pending the push translations are polled for processing. Default value is 00:15:00.000.
updateLeasedByPerNumberOfItems	Number of items that have to be updated before the translation job is updated. Default value is 100.

Application configuration for TranslationOrganizer

Modify the `TranslationOrganizer.exe.config` file parameters noted below to configure or customize TranslationOrganizer.

The file is located on the Content Manager server: `\InfoShare\App\TranslationOrganizer\Bin\TranslationOrganizer.exe.config`.

The parameters within the section **trisoft.infoShare.translationOrganizer** are grouped as follows:

- The `Settings` element contains the common parameters. They are all related to the working of the TranslationOrganizer service or with the update of the Content Manager Repository.
- The `worldServer/instances/add` element contains the parameters specific to SDL WorldServer.
- The `tms/instances/add` element contains the parameters specific to SDL TMS.
- The `fileSystem/instances/add` element contains the parameters specific to the File System.

TranslationOrganizer common parameters

TranslationOrganizer common parameters are grouped in the `Settings` element of the `trisoft.infoShare.translationOrganizer` section of the `\InfoShare\App\TranslationOrganizer\Bin\TranslationOrganizer.exe.config` file.

dumpFolder

The folder where the temporary files are created. This value is initialized from install parameters.

maxTranslationJobItemsUpdatedInOneCall

Maximum number of items updated in a single transaction. Default value is 100.

jobPollingInterval

Interval at which jobs are polled for processing. Default value is 00:05:00.000.

pendingJobPollingInterval

Interval at which jobs pending translation are polled for processing. Default value is 00:15:00.000.

systemTaskInterval

The minimal interval that system tasks (for example, template synchronisation) are run. Default value is 00:10:00.000.

attemptsBeforeFailOnRetrieval

Number of attempts the update of single content object fails before the job is moved to failed status. The default value is 3.

updateLeasedByPerNumberOfItems

Number of items that have to be updated before the translation job is updated. Default value is 100.

synchronizeTemplates

Specifies whether service should synchronize templates.

retriesOnTimeout

Number of times the single server call can fail and be retried before the job is moved to failed status. Default value is 3.

TranslationOrganizer WorldServer parameters

TranslationOrganizer WorldServer parameters are grouped in the `worldServer/instances/` add element of the `trisoft.infoShare.translationOrganizer` section of the `\InfoShare\App\TranslationOrganizer\Bin\TranslationOrganizer.exe.config` file.

alias

An unique display name for the SDL WorldServer installation (e.g. `wsDemo`, `prod`, `dev`,...).

uri

The base URI for SDL WorldServer (e.g. `http://worldserver.example.com:8080/ws-legacy/services` when using legacy web services or `http://worldserver.example.com:8080/ws-api` when using the new REST API). This value has to be set manually.

userName

The user name to access SDL WorldServer. This value has to be set manually.

password

The password to access SDL WorldServer. This value has to be set manually.

externalJobMaxTotalUncompressedSizeBytes

Maximum total size of the single translation job. Default value is 5242880 bytes.

retriesOnTimeout

Number of times the single external call can fail and be retried before the job is moved to failed status. Default value is 3.

apiProtocol

The protocol that is used to connect to SDL WorldServer. Use `SOAP` to connect using the legacy web services. Use `REST` to connect using the latest REST API. Default value is `REST`.

httpTimeout

The timeout used when doing the WorldServer call using REST API. Default value is 00:02:00.000.

The language mapping

The configuration must also contain a mapping of the language from Content Manager to the locale of SDL WorldServer.

```
<mappings>
<add trisoftLanguage="en" worldServerLocaleId="1145" />
<add trisoftLanguage="nl" worldServerLocaleId="1147" />
<add trisoftLanguage="fr" worldServerLocaleId="1146" />
</mappings>
```

Note: No more than one SDL WorldServer installation instance is supported in this release. If you are not configuring for use with SDL WorldServer, do not configure any instances for SDL WorldServer.

TranslationOrganizer TMS parameters

TranslationOrganizer TMS parameters are grouped in the `tms/instances/add` element of the `trisoft.infoShare.translationOrganizer` section of the `\Infoshare\App\TranslationOrganizer\Bin\TranslationOrganizer.exe.config` file.

alias

A unique display name for the SDL TMS installation (e.g. demo, prod, dev,...).

uri

The base URI for SDL TMS (e.g. `http://tms.example.com/`). A value is mandatory.

Authentication parameters

Depends on the `authenticationMode` setting: the type of authentication used for signing in SDL TMS from Content Manager. Possible values: `ApiSecret` (the SDL TMS legacy authentication mode), `UserCredentials` (the current SDL TMS authentication mode). The default value is `ApiSecret`.

ApiSecret: legacy authentication mode (SDL TMS 12 and earlier)

- **apiKey:** the API key that is associated with a user (is used only by REST API). It can be obtained from TMS (**Setup > Users & Groups > Users > "<user name>" > TMS API key**). A value is mandatory.
- **secret:** API key encrypted using the user password (is used only by REST API). You can obtain the secret by contacting your SDL TMS contacts or contact SDL Support and providing them the API key as well as the username and password of the user performing the integration. A value is mandatory.

UserCredentials: current SSO authentication (SDL TMS 12.1 and later)

- **Username:** the `username` of the user who is signing in.
- **Password:** the password of the user who is signing in.

externalJobMaxTotalUncompressedSizeBytes

Maximum total size of the single translation job. Default value is 5242880 bytes.

retriesOnTimeout

Number of times the single external call can fail and be retried before the job is moved to failed status. Default value is 3.

httpTimeout

The timeout used when doing the SDL TMS call using REST API. Default value is 00:02:00.000.

destinationPortNumber

Gets or sets a value indicating the TCP Port number over which communications should be conducted.

isapiFilterLocation

Gets or sets a value indicating the path of the CTAISAPI component in SDL TMS that will receive communication requests.

useCompression

Gets or sets a value indicating whether communications should be compressed (is used only by CTAISAPI).

useSsl

Gets or sets a value indicating whether communications should be conducted over a secure channel (is used only by CTAISAPI). If the URI is `https`, this parameter must be set to `true`.

useDefaultProxyCredentials

Gets or sets a value indicating whether to use default credentials when communicating with a Proxy server.

proxyServer

Gets or sets a value indicating the URI of the Proxy server to use.

proxyPort

Gets or sets a value indicating the TCP port to use when communicating with the proxy.

The language mapping

The configuration must also contain a mapping of the language from Content Manager to the language of SDL TMS:

```
<mappings>
<add trisoftLanguage="en" tmsLanguage="EN" />
<add trisoftLanguage="fr" tmsLanguage="FR" />
<add trisoftLanguage="de" tmsLanguage="DE" />
<add trisoftLanguage="nl" tmsLanguage="NL" />
</mappings>
```

The SDL TMS configurations that specify the workflow and the language pairs. Each configuration that you want to use within Content Manager must be configured within a `<template>` element:

```
<templates>
<add templateId="81143C38-0C96-4A8C-9BBB-87C1CF464FE3"
templateName="My template" />
<add templateId="70407FBC-86FA-4A9D-8E6D-35E1AE85DB73"
templateName="Trisoft template" />
</templates>
```

Note: After selecting a `template`, only the target languages that are configured for that SDL TMS configuration can be used as possible target language within Content Manager.

Metadata settings

The configuration can contain:

- The metadata that will be extracted and passed to SDL TMS.

```
<requestedMetadata>
<ishfields>
<ishfield name="FAUTHOR" level="lng" ishvaluetype="value" />
<ishfield name="DOC-LANGUAGE" level="lng" ishvaluetype="value" />
</ishfields>
</requestedMetadata>
```

- The metadata that will be used for grouping items in SDL TMS.

```
<groupingMetadata>
<ishfields>
<ishfield name="FAUTHOR" level="lng" ishvaluetype="value" />
<ishfield name="DOC-LANGUAGE" level="lng" ishvaluetype="value" />
</ishfields>
</groupingMetadata>
```

This metadata will be passed to SDL TMS together with the metadata specified in the `requested-Metadata` section.

Note: No more than one SDL TMS installation instance is supported in this release. If you are not configuring for use with SDL TMS, do not configure any instances for SDL TMS.

TranslationOrganizer File System settings

TranslationOrganizer file system parameters are grouped in the `fileSystem/instances/add` element of the `trisoft.infoShare.translationOrganizer` section of the `\InfoShare\App\TranslationOrganizer\Bin\TranslationOrganizer.exe.config` file.

alias

An unique display name for the configured File System (e.g. demo, prod, dev,...).

externalJobMaxTotalUncompressedSizeBytes

Maximum total size of the single translation job. Default value is 5242880 bytes.

exportFolder

Export folder wherer the exported zip archives will be stored. This value is initialized from install parameters.

Metadata settings

The configuration can contain the metadata that will be extracted and exported as `.met` files next to the actual files.

```
<requestedMetadata>
  <ishfields>
    <ishfield name="FAUTHOR" level="lng" ishvaluetype="value" />
    <ishfield name="DOC-LANGUAGE" level="lng" ishvaluetype="value" />
  </ishfields>
</requestedMetadata>
```

Note: No more than one File System instance is supported in this release. If you are not configuring for use with File System, do not configure any instances for File System.

Translation Job Workflow

The life cycle of the translation job.



The main workflow of the translation job.



Failing steps of the translation job workflow.

- TranslationBuilder is responsible for unlocking outdated leases of TranslationJobs so they are part of the workflow again.
- Manual actions can only be executed on TranslationJobs which are not having a status controlled by the two services. This includes Definition, Completed, Cancelled and all Failed statuses.

Integration requirements for Content Manager and SDL WorldServer

The following will help you better understand the integration of Content Manager and SDL WorldServer and considerations when configuring translation and workflow.

Content Manager

Required for Content Manager integration:

- Languages are defined as described for source and pivot languages (see related topic). For example: The following example defines a source language of English with target languages for German, Spanish, French, Italian and Chinese. The last group indicates that Chinese can be used as source language (pivot) for translating to Japanese and Korean.

```
<languagepaths>
<languagepath from="nl" to="fr"/>
<languagepath from="en" to="de"/>
<languagepath from="en" to="es"/>
<languagepath from="en" to="fr"/>
<languagepath from="en" to="it"/>
<languagepath from="en" to="zh"/>
<languagepath from="zh" to="ja"/>
<languagepath from="zh" to="ko"/>
</languagepaths>
```

- Translation templates come from SDL WorldServer and are automatically pushed to Content Manager by TranslationOrganizer.
- Content Manager uses UTF-16 for the content.
- The dedicated user (typically *ServiceUser*) having the role TRANSLATORSERVICE.
- Statuses and status transitions, configured for the TRANSLATORSERVICE role and for user roles, that defines a workflow for the integration:
 - There is an initial status to indicate that the object is ready for translation: **To be translated**. This status is used by TranslationBuilder to create new target language objects and by TranslationOrganizer to identify and include objects to send.
 - With the status transition from the initial status, there is a status to indicate that the object is no longer under control of the CMS: **In translation**.
Status transition: **To be translated** to **In translation**.

TranslationJob option `Include 'In translation' items` forces the re-sending of objects in this status.

- With the status transition from the status above, there is a status to indicate that the object is back in CMS control: **Translation in review**.

Status transition: **In translation** to **Translation in review**.

- With a status transition from the **Translation in review**, there is a status indicating that a user approved the translation: **Translation approved**.

Status transition: **Translation in review** to **Translation approved**.

- With a status transition from the **Translation in review**, there is a status indicating that a user rejected the translation and the object needs to be re-translated: **Translation rejected**.

Status transition: **Translation in review** to **Translation rejected**.

- Status transitions allow for objects to be re-translated from several positions in the flow, thus making the process cancelled or rolled back to the status **To be translated**.

Status transitions:

- **In translation** to **To be translated**.
- **Translation in review** to **To be translated**.
- **Translation rejected** to **To be translated**.

The TRANSLATORSERVICE role and the status transitions are configured accordingly out-of-the-box.

SDL WorldServer

Required for SDL WorldServer integration:

Note: Refer to the SDL WorldServer manuals for the further details about how to configure the workflow in SDL WorldServer, and related settings.

- A dedicated user in SDL WorldServer.
This is the user that is used by Translation Organizer to logon to SDL WorldServer. This is the parameter `userName` within the `worldServer` section in the configuration file.
- Content Manager mount configured to use UTF-16.
- Locales, Workflow, Project Types, whatever is required by WorldServer to go through the translation process.
- A first step named *Translate*.
- After the *Translate* step there should be an automatic *Save* step.
- After the *Save* step there should be *Translated Content Retrieval* step which is used by Content Manager to start getting the translation.
- After the *Translated Content Retrieval* step, a transition named **Ready for review** leads to the *Translated Content In Review* step.
- After the *Translated Content In Review* step, there must be two possible transitions:
 - The **Approved** transition that leads to the end of the flow.
 - The **Rejected** transition that leads back to the *Translate* step.
- The *Asset path normalizer*. This optional custom component can be implemented separately and

installed in SDL WorldServer to force SDL WorldServer TM to consider the file name when doing the match.

Compatibility

- For backward compatibility, it is still possible to configure a simplified flow made of *Translated Content Retrieval* immediately followed by a transition called **Retrieved**.
- The translation review flow with the possibility to approve or reject the translation, introduced in Content Manager 13.0.0, is incompatible with the SOAP API protocol. Therefore when WorldServer uses that protocol, only the legacy flow can be used (*Translated Content Retrieval* immediately followed by **Retrieved**).
- Any given job must be using only one type of flow: Either the current flow with an approval step, or the legacy flow. The system can not manage mixed flows.

Integration requirements for Content Manager and SDL TMS

The following will help you better understand the integration of Content Manager and SDL TMS and how to configure translation and workflow.

Content Manager

Required for Content Manager integration:

- Languages are defined as described for source and pivot languages (see related topic). For example: The following example defines a source language of English with target languages for German, Spanish, French, Italian and Chinese. The last group indicates that Chinese can be used as source language (pivot) for translating to Japanese and Korean.

```
<languagepaths>
<languagepath from="nl" to="fr"/>
<languagepath from="en" to="de"/>
<languagepath from="en" to="es"/>
<languagepath from="en" to="fr"/>
<languagepath from="en" to="it"/>
<languagepath from="en" to="zh"/>
<languagepath from="zh" to="ja"/>
<languagepath from="zh" to="ko"/>
</languagepaths>
```

- Translation templates come from SDL TMS and are automatically pushed to Content Manager by TranslationOrganizer.
- Content Manager uses UTF-16 for the content.
- The dedicated user (typically *ServiceUser*) must have the role TRANSLATORSERVICE.
- Statuses and status transitions, configured for the TRANSLATORSERVICE role and for user roles, define a workflow for the integration:
 - There is an initial status to indicate that the object is ready for translation: **To be translated**. This status is used by TranslationBuilder to create new target language objects and by TranslationOrganizer to identify and include objects to send.
 - With the status transition from the initial status, there is a status to indicate that the object is no longer under control of the CMS: **In translation**.
Status transition: **To be translated** to **In translation**.

TranslationJob option `Include 'In translation' items` forces the re-sending of objects in this status.

- With the status transition from the status above, there is a status to indicate that the object is back in CMS control: **Translation in review**.

Status transition: **In translation** to **Translation in review**.

- With a status transition from the **Translation in review**, there is a status indicating that a user approved the translation: **Translation approved**.

Status transition: **Translation in review** to **Translation approved**.

- With a status transition from the **Translation in review**, there is a status indicating that a user rejected the translation and the object needs to be re-translated: **Translation rejected**.

Status transition: **Translation in review** to **Translation rejected**.

- Status transitions allow for objects to be re-translated from several positions in the flow, thus making the process cancelled or rolled back to the status **To be translated**.

Status transitions:

- **In translation** to **To be translated**.
- **Translation in review** to **To be translated**.
- **Translation rejected** to **To be translated**.

The TRANSLATORSERVICE role and the status transitions are configured accordingly out-of-the-box.

SDL TMS

Required for SDL TMS integration:

- Configurations specifying the workflow and all necessary language pairs.

Remember: When a configuration is selected, only the target languages (and workflow) that are configured for the source language can be used.

- The workflow should contain minimally the following steps in TMS:
 - The *Translation* step. In TMS by default.
 - The *Translated Content retrieval* step which is used by Content Manager to start getting the translation. In TMS by default.
 - The *Translated Content In Review* step. You need to add this step manually, as a human step (it is not performed by the system).
- Every step between *Translation* (not included) and *Translated Content In Review* (included) must have the user performing the integration assigned as default user. If it is not, then the integration user must be included in every group the steps are assigned to.
- Refer to the SDL TMS manuals for the further details about how to configure the workflow in SDL TMS.

Compatibility

- The oldest required SDL TMS version allowing the use of **Translation rejected** along with all the cancellation options is SDL TMS 11.2.1.
- Any given job must be using only one type of flow: Either the current flow with an approval step, or the legacy flow. The system can not manage mixed flows.

Integration requirements for Content Manager and File System

The following are items that will help you better understand the integration of Content Manager and File System and considerations when configuring translation and workflow.

Content Manager

Required for Content Manager integration:

- Languages are defined as described for source and pivot languages (see related topic). For example: The following example defines a source language of English with target languages for German, Spanish, French, Italian and Chinese. The last group indicates that Chinese can be used as source language (pivot) for translating to Japanese and Korean.

```
<languagepaths>
<languagepath from="nl" to="fr"/>
<languagepath from="en" to="de"/>
<languagepath from="en" to="es"/>
<languagepath from="en" to="fr"/>
<languagepath from="en" to="it"/>
<languagepath from="en" to="zh"/>
<languagepath from="zh" to="ja"/>
<languagepath from="zh" to="ko"/>
</languagepaths>
```

- Translation templates are based on configured source and pivot languages, and are automatically pushed to Content Manager by TranslationOrganizer.
- Content Manager uses UTF-16 for the content.
- The dedicated user (typically *ServiceUser*) having the role TRANSLATORSERVICE.
- Statuses and status transitions, configured for the TRANSLATORSERVICE role, that defines a workflow for the integration:
 - There is an initial status to indicate that the object is ready for translation: **To be translated**. This status is used by TranslationBuilder to create new target language objects and by TranslationOrganizer to identify and include objects to send.
 - With the status transition from the initial status, there is a status to indicate that the object is no longer under control of the CMS: **In translation**.
Status transition: **To be translated** to **In translation**.
TranslationJob option `Include 'In translation' items` forces the re-sending of objects in this status.
 - With the status transition from the status above, there is a status to indicate that the object is back in CMS control: **Translation in review**.
Status transition: **In translation** to **Translation in review**.
 - With a status transition from the **Translation in review**, there is a status indicating that a user approved the translation: **Translation approved**.
Status transition: **Translation in review** to **Translation approved**.
 - With a status transition from the **Translation in review**, there is a status indicating that a user rejected the translation and the object needs to be re-translated: **Translation rejected**.
Status transition: **Translation in review** to **Translation rejected**.
 - Status transitions allow for objects to be re-translated from several positions in the flow, thus making the process cancelled or rolled back to the status **To be translated**.

Status transitions:

- **In translation** to **To be translated**.
- **Translation in review** to **To be translated**.
- **Translation rejected** to **To be translated**.

The TRANSLATORSERVICE role and the status transitions are configured accordingly out-of-the-box.

File System

Required for File System integration:

- The dedicated user (see above) configured for the integration has an access to the folder where the zip files should be created.

Enabling services

After the install is complete, services will not start automatically, since the database is not guaranteed to be in the right state until you run DBUT tool. Also, you might decide not to start some services on the specific installation depending on the server role. To enable typical services you can locate and run the Enable-DefaultServices.ps1 script.

Before you begin

- DBUT completed successfully.
- The Administrator setup completed successfully.
- The TRANSLATORSERVICE role is created and the translation status transitions have been assigned to it.
- System Administrator rights.

Procedure

1. Run the script `\App\Setup\Manage\Enable-DefaultServices.ps1`
Typical services (Trisoft InfoShare Crawler One, Trisoft InfoShare SolrLucene, Trisoft InfoShare BackgroundTask One, Trisoft InfoShare TranslationBuilder One etc.) are started, startup type is set to "Automatic (Delayed Start)".

Enabling Draft Space

We recommend you enable Draft Space with ISHDeploy.

Before you begin

Check how to access to and use ISHDeploy in the documentation located [here](#).

About this task

Draft Space is one of two tools that are grouped under the name Collective Spaces.

Procedure

1. Enter the following command:

```
Enable-ISHUICollectiveSpaces [-DraftSpace] [-ISHDeployment <ISHDeployment>]
```

Results

You now have access to Draft Space for web browser-based content authoring.

Rebuilding the full text index

You can rebuild the full text index if it is no longer synchronized with the current database, or after a Content Manager upgrade on a new server for example.

Before you begin

- You must be a member of the system administration user group.
- You must have full access to the application server.
- The crawler service must be running.

Note: We recommend that you rebuild the index outside regular business hours as the initialization uses database resources. In addition, the index may take a while to rebuild depending on the size of the repository.

Procedure

1. On the application server, open the \InfoShare\App<projectsuffix>\Crawler\Configuration folder.
2. Execute the StartDataFolderCleanup.bat file as administrator, to remove the existing index.
3. Execute the StartReindex.bat file as administrator, to rebuild it based on the existing repository and project.

Configuring Content Manager for Collaborative Review

To enable accessing Collaborative Review features from Content Manager, you need to specify values including the URI of your Collaborative Review web application in two different configuration settings.

Specify the Collaborative Review URI in Content Manager

To access Collaborative Review features from Content Manager, you need to specify the Collaborative Review URI in the Content Manager **Settings**.

Procedure

1. Log on to the SDL Tridion Docs Web Client as a user with administrative permissions.
2. Select the slide-out navigation pane, then Content Manager.
3. Specify the Collaborative Review URI:
 - a. On the **Settings** tab, select **Default Settings**.
 - b. Under **Collaborative Review URI**, specify the web address of the Collaborative Review web application:

`https://delivery.lc.example.com/CollaborativeReview/` (where `lc` refers to an example related to output, and `delivery` specifies it further as a delivery server)

Modify the SDL Tridion Docs output type settings

Follow these steps if you want to make Content Manager able to publish to Collaborative Review.

Procedure

1. Log on to the SDL Tridion Docs Web Client as a user with administrative permissions.
2. Select the slide-out navigation pane, then Content Manager.
3. On the **Settings** tab, select **Output Formats**.
4. Modify the **Content Manager** output type to specify the Collaborative Review web application settings.
5. Select the **Content Delivery** output type and select **Properties**.
The **Output Format Properties** window pops up.
6. Ensure certain fields are correctly filled or cleared for the **Content Delivery** output type.
 - a. In the **Advanced** Section, ensure that **Resolve Variables** is cleared.
 - b. In the Collaborative Review section, specify details for Collaborative Review web application in the following fields.

URI of the server

The fully qualified Collaborative Review server name, in this format:

`https://delivery.lc.example.com/CollaborativeReview/` (where `lc` refers to an example related to output, and `delivery` specifies it further as a delivery server)

Skin used

The skin to use for the **Content Delivery** output type.

User name

Provided in combination with the password for a user who has at least edit privileges in SDL Tridion Docs

Note: If you are using ADFS for SSO, leave the **User name** and **Password** fields blank.

Password

Provided in combination with the password for a user who has at least edit privileges in SDL Tridion Docs

Note: If you are using ADFS for SSO, leave the **User name** and **Password** fields blank.

Verifying the installation

Follow these procedures to test and verify the critical parts of the new Content Manager installation.

About this task

In general the following should never result in an error and cause no harm on a production system. Testing more complex setups, such as the batch servers is out-of-scope.

Procedure

1. Log in into the Content Manager Web client as an existing user.
For example, go to **http://servername/InfoShareAuthor** and login using an existing login name and password, or the admin name and password.
2. Test the inbox by selecting the **Inbox** tab then select one of the inbox tabs in the left pane.
Test is successful if the inbox is displayed in the right pane in a table format.
3. Create a folder in the repository:
 - a. Select the **Repository** tab.
 - b. Create a folder by clicking the **New Folder** icon in the upper left pane.
 - c. Enter a name.
 - d. Click **Ok**.

Test is successful if the folder is created (shown in the left pane). Remove the folder after verifying using the **Delete Folder** button in the upper left pane.
4. Test publishing; if a publication is not available, skip this test.
 - a. Select the **Repository** tab.
 - b. Locate a publication (traverse the left pane) and select the publication (in the right pane).
 - c. Select an output type in the bottom pane.
 - d. Click **Publish**.
A confirmation window is displayed.
 - e. Click **OK** in the confirmation window.
The publication output status (bottom right pane of the search window) changes to **Pending**.
Test is successful if the output status changes to **Draft**; refresh the view to confirm that the status changes to **Draft**.
5. Execute a full text search:
 - a. Select the **Search** tab.
 - b. Enter a term in the **Search term** field.
Be sure to enter a word that you know exists in your topics. For example, enter the search word: **the**.

- c. Click **Search**.

Test is successful if search results are displayed in the right pane.

6. Verify web services, synchronization, and network availability by starting the Client Tools such as Publication Manager, Condition Manager or Authoring Bridge.

Test is successful if you can view the repository through the client tools and can view or preview a topic in the repository.

7. Verify requested customizations to your system such as PDF stylesheets, extra metadata, or extra development for integration with other systems (such as SDL-TMS, PLMs, SingleSignOns).

Ask your Content Manager administrator or documentation manager what customizations, if any, were requested.

Verifying URLs

If your environment includes reverse proxy servers, after you install Content Manager, you need to check that certain URLs are accessible.

Procedure

- Using Windows Internet Explorer, ensure that you can reach the following URLs:
 - `BaseURL/ISHWS/Application.asmx?wsdl`
 - `BaseURL/ISHWS/Application.asmx?disco` where *BaseURL* is the value specified for the `baseurl` parameter, and *ISHWS* is the value specified for the `infosharewebappname` parameter. These parameters are set in the `inputparameter.xml` file that is used by the Content Manager installer.
- If you cannot access the URLs, verify that the reverse proxy servers are correctly configured. See the documentation for the reverse proxy servers.

Verify read access to the database by viewing an inbox

To verify read access to the database, request to view an inbox in the web client.

Procedure

- Open Internet Explorer and enter the address for the Content Manager web client.

Note: The web client address is a combination of the value in the parameters in the `inputparameters.xml` file:

`baseurl/infoshareauthorcmwebappname/`

For example, if:

```
<param name="baseurl">
<currentvalue>https://example.corp</currentvalue>
```

and

```
<param name="infoshareauthorwebappname">
<currentvalue>ISHCM</currentvalue>
```

then the URL is:

<https://example.corp/ISHCM/>

Enter a Content Manager username and password. If you are not sure about the login/password and you imported the default database you can use **admin/admin** to login.

2. Access the SDL product menu by clicking on the icon in the upper left of the window and select **Content Explorer**.
3. If not selected, select the **Inbox** tab at the top of the window.
4. In the left pane, select one of the inboxes, for instance, select **Reviewer**.
If no objects are in the inbox, an empty inbox is displayed; **No objects in inbox** is reported in the right pane.
If there are objects in the inbox, a list of objects is displayed in the right pane.

Verify read and write access to the database by creating a folder

Create, modify, and delete actions are handled through transactions and verifies read and write access to the database.

Procedure

1. Login to the Content Manager web client as an existing user. For example as the admin user.
2. Select the **Repository** tab at the top of the window.
3. Click the **New Folder** icon in the upper left of the Repository pane.
A **New Folder** window displays.
4. Enter a name in the **Folder Name** field. For example enter the folder name, **Test**.
5. Click **OK**.
The folder is created and displayed in the left pane.

What to do next

You can remove the test folder by selecting it in the left pane then clicking the **Delete Folder** icon (red X) in the upper left pane. You are asked to confirm the delete action, click **Yes** to confirm and delete the folder.

Creating an account and connecting to the Repository

Before you can use Content Importer, you need to set up your user account. If you installed Content Importer on a system where Publication Manager is already installed, the user account settings are recognized.

Before you begin

You must create the user account and configure the connection to the Repository to allow a user to connect and access data in the Content Manager Repository. Use this procedure to create a new user account for testing purposes.

To create the account and connect to the Repository, you need the URL of the Content Manager web services.

Procedure

1. In Content Importer, select **Create an account**.
The account window is displayed.
2. Enter an **Account Name** and the URL of the Content Manager **Web Service**.
3. Select **Next**.
4. If necessary, select the **Authentication Method**.
5. Enter the username and password.
Check **Remember password** if you do not want to enter the information each time you use the application.
6. Select **Next**.
Content Manager validates the account and synchronizes files.

Running a client tool

When started, the client tools verify availability of the synchronization website and web services.

Before you begin

- A desktop client workstation must be installed with the client tools. If not done, refer to the section for installing desktop clients.
- The client tool must be configured with a user login and account.
- To fully test the client, the database should contain data.

Procedure

1. If necessary, create an account and connection to the repository.
2. Start a client tool such as Publication Manager, Condition Manager or Authoring Bridge from the **Start** menu or desktop shortcut.

If you can view and access the repository through the client tool, and can view or preview a topic in the repository then web services, synchronization, and network availability have been successfully verified.

Testing publishing

You can test the publish functionality if your database contains topics, maps and publications, and it is configured to render an output type.

Before you begin

If your database is not configured to render an output, refer to the [Content Manager Information Portal](#) documentation for information about adding output formats.

Procedure

1. Login to the Content Manager web client as an existing user. For example as the admin user.
2. Select the **Repository** tab at the top of the window.
3. Select a publication in the left pane.
4. Select a version in the upper right pane by clicking on the publication name.
The available versions of the publication are listed in the lower right pane.
5. Select a version and language for the publication in the bottom right pane.
This is done in the right and left check boxes next to the output format.
6. Click the **Publish** button in the menu on the bottom right.
A confirmation dialog displays.
7. Click **OK** to confirm and begin the publish process.
8. To display progress, click **Refresh**.
Verification is complete when the Event Description is **Publish Process ended** and the status says **SUCCESS**.
9. Click **Close**.

Executing the full text search

You can test the search functionality if the database contains content.

Procedure

1. Login to the Content Manager web client as an existing user. For example as the admin user.
2. Select the **Search** tab at the top of the window.
3. Enter a word in the **Search Term** field then click **Search**.
If there is no search result, verify if a rebuild of the full-text-index collection is required. Note that the full-text-index collection is not immediately available after installation since it takes some time to build.

Verifying customer specific components

If you requested customizations of the out-of-the-box Content Manager software, a check that they were delivered is recommended at this time.

About this task

Customizations of the delivered software may be, for example, PDF stylesheets, extra metadata, or extra development to integrate with other systems such as SDL-TMS, PLMs, SingleSignOns.

If you contracted for additional customizations, check that they were delivered and functioning as required.

Collaborative Review upgrade

Collaborative Review has added features.

If you were using an earlier version of Collaborative Review and just upgraded to the latest one, you must republish your documents to your Collaborative Review server to use the new review and edit features of the product.

System management reference for Content Manager upgrade

A few system management concepts are good to know in the context of a complex Content Manager upgrade.

Introduction to SDL Tridion Docs scalability

A number of deployment scenarios offers an overview of different scaling out options.

Scalability is the ability of a system, network, or process to handle a growing amount of work in a capable manner or its ability to be expanded to accommodate that growth.

This introduction presents:

- The high-level description of a cluster that would involve all components of Tridion Docs, and the variations caused by the use of Single Sign-On (SSO) solutions, such as ISHSTS or ADFS.
- Levels of scaling out with Content Manager.

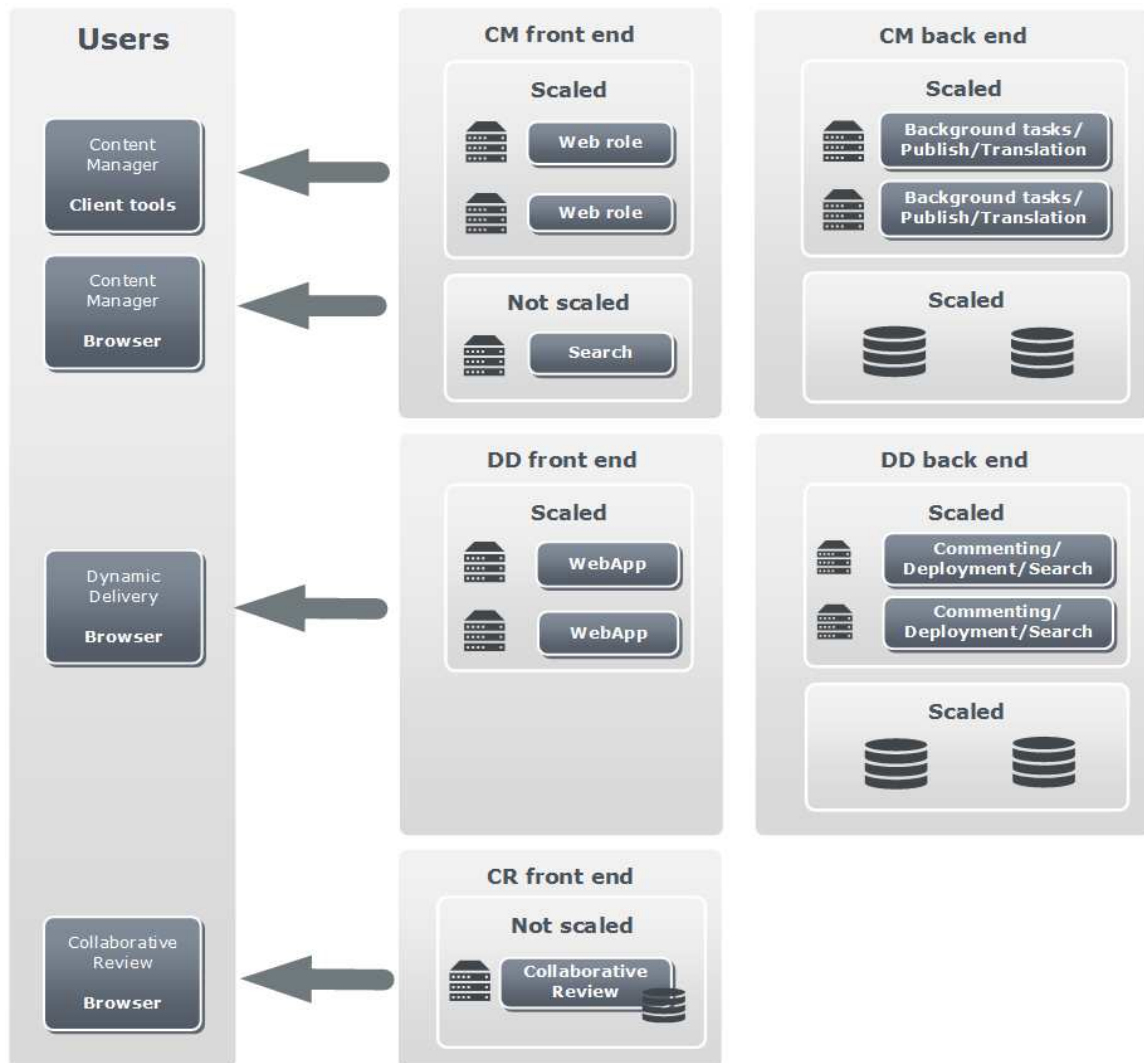
Some capabilities require affinity and some don't. The affinity is the fact that a request would go to a preferred destination server depending on the sender, whereas normally it would just go to the next available destination.

- Content Manager doesn't require affinity.

SDL Tridion Docs network cluster

Example of a combined Content Manager, Dynamic Delivery and Collaborative Review network cluster setup.

The following diagram shows how Content Manager and Dynamic Delivery can be scaled up.



Related concepts

- “Content Manager scalability” on page 60

SDL Tridion Docs environment with ISHSTS

Example of a combined Content Manager, Dynamic Delivery and Collaborative Review network setup with ISHSTS integration.

The following clusters can fit in one SDL Tridion Docs deployment:

- Content Manager advanced server cluster.
- Dynamic Delivery advanced server cluster.
- Collaborative Review cluster.

Content Manager

A collection of front end servers behind a network load balancer serve interactive functionality, whereas a collection of back end servers serve non-interactive functionality.

ISHSTS is a Security Token Service that is part of the Web role.

When designing a cluster like the above you should take special notice for the following items.

- Each Front end server behind the network load balancer is configured using the same certificate referring to the same host name.
- Every Back end server should be installed with its own certificate referring to its unique host name.
- For every federated service endpoint e.g. ISHWS, targeted from within the cluster, DNS resolving and network routing should be taken into consideration depending on the network topology.
- ISHSTS cannot be shared across different servers. As a result:
 - Every ISHSTS on every server on the cluster requires configuration for all federated services for which it can potentially issue a token.
 - ISHSTS on every Front end server has configuration based on the network load balancer hostname and certificate. Also it must have all required configuration relevant to other federate services as their endpoints are recognized from outside the cluster.
 - ISHSTS on every Back end server has configuration based on the specific hostname and certificate of the server. This ISHSTS will be used by all entities of the same Back end server. All federated services integrated with Content Manager are required to be configured on the ISHSTS on every Back end server, using endpoints relevant to configured DNS resolving and network routing.

With a setup similar to this all user clients like browsers and client tools will target the network load balancing hostname and thus one of the Front end server. Any client that is running from within the cluster behind the network load balance will still have access to any Back end server by using its designated host name.

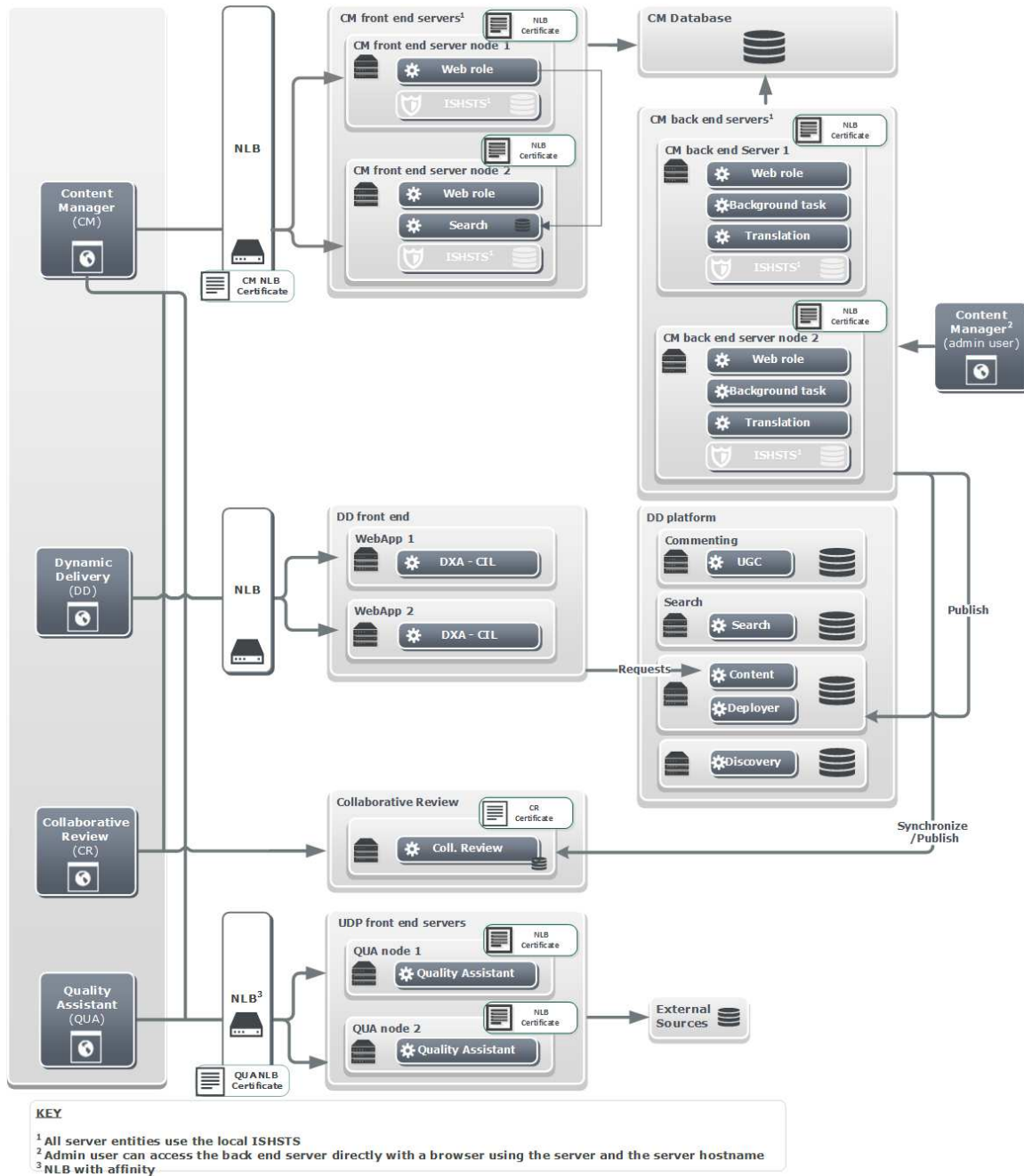
Dynamic Delivery

The delivery servers can be scaled out behind a network load balancer. Commenting and search are provided from Back end services.

A deployment node is the target of publications from Content Manager. This node will be queried by each delivery server in the cluster.

Collaborative Review

The review installation provides the functionality for Collaborative Review. The source of the comments is the common commenting repository. This installation is integrated with ISHSTS to provide the Single Sign On experience.



SDL Tridion Docs environment with ADFS

Example of a combined Content Manager, Dynamic Delivery and Collaborative Review network setup with ADFS integration.

The following clusters can fit in one SDL Tridion Docs deployment:

- Content Manager advanced server cluster.
- Dynamic Delivery advanced server cluster.
- Collaborative Review cluster.

Federation

ADFS is used as a Security Token Service for the federated services of Content Manager, Dynamic Delivery and Collaborative Review.

Content Manager

A collection of Front end servers behind a network load balancer serve the interactive functionality and a collection of Back end servers serve the non interactive functionality

When designing a cluster like the above you should take special notice for the following items.

- Each Front end server behind the network load balancer is configured using the same certificate referring to the same host name.
- Every Back end server should be installed with its own certificate referring to its unique host name.
- For every federated service endpoint e.g. Content Manager ISHWS, targeted from within the cluster, DNS resolving and network routing should be taken into consideration depending on the network topology.

With a setup similar to this all user clients like browsers and client tools will target the network load balancing hostname and thus one of the Front end server. Any client that is running from within the cluster behind the network load balance will still have access to any Back end server by using its designated host name.

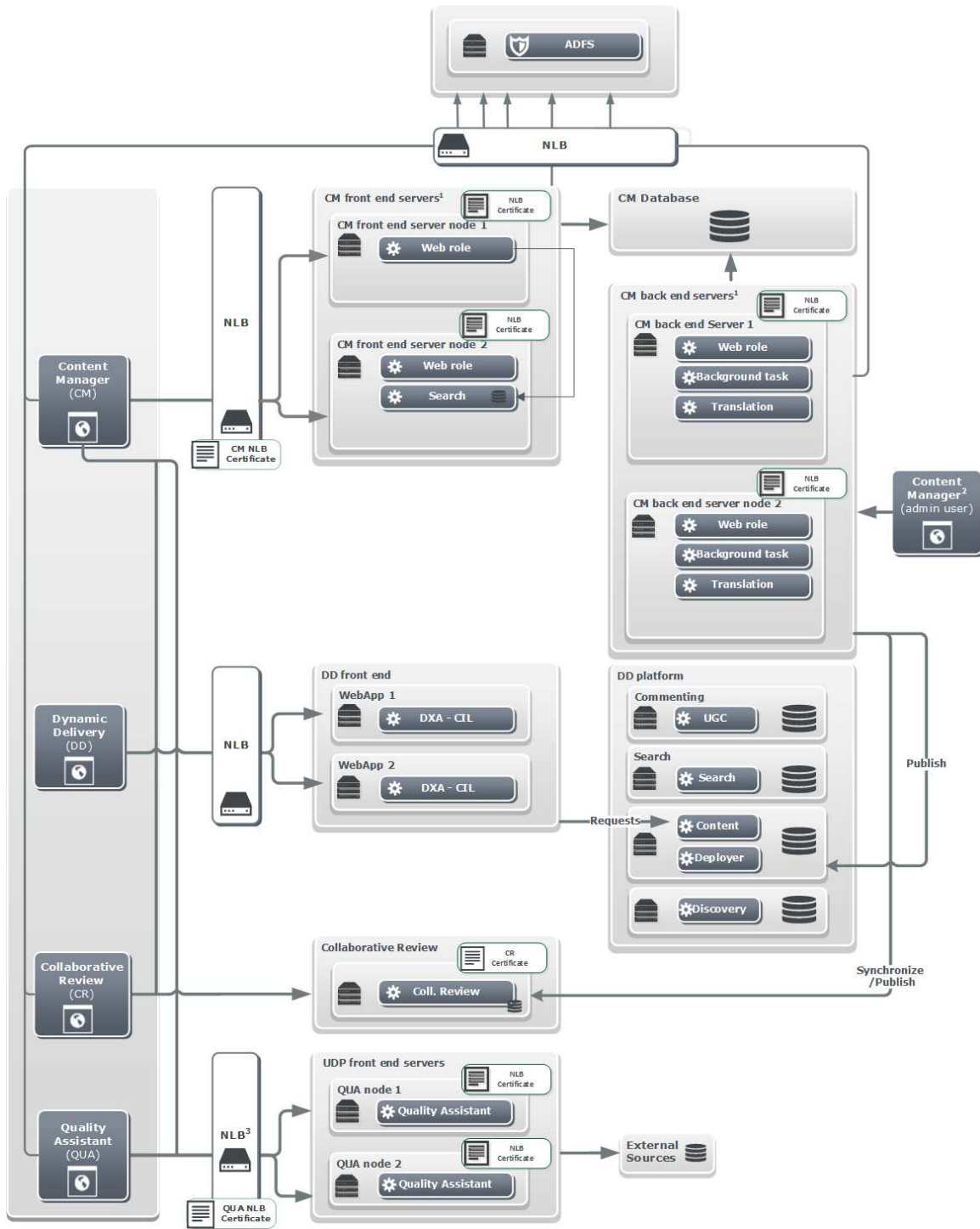
Dynamic Delivery

The delivery servers can be scaled out behind a network load balancer. Commenting and search are provided from Back end services.

A deployment node is the target of publications from Content Manager. This node will be queried by each delivery server in the cluster.

Collaborative Review

The review installation provides the functionality for Collaborative Review. The source of the comments is the common commenting repository. This installation is integrated with ADFS to provide the Single Sign On experience.



KEY

¹ All server entities use the local ISHSTS

² Admin user can access the back end server directly with a browser using the server and the server hostname

³ NLB with affinity

Content Manager scalability

There are many ways to scale up your Content Manager network as your content grows.

Related concepts

- “SDL Tridion Docs network cluster” on page 54

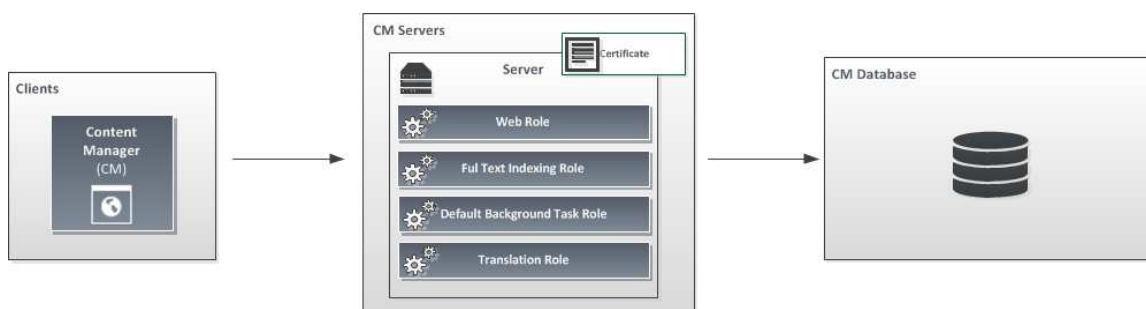
One server for all roles

The basic deployment option consists of one server that is responsible for all roles.

For Content Manager to operate correctly, all server roles need to be active:

1. The Default background task role provides execution for the background.
2. The Full text indexing role provides crawling and indexing functionality for the SolrLucene search engine.
3. The Translation role provides all translation-related functionality.
4. The Web role provides all necessary web endpoints like web site, web services, and the internal security token service.

The following diagram shows a conceptual representation of the one server deployment.



All these roles work together to provide the expected functionality. Depending on the load and the intensity of usage, the execution of the above roles might not work at peak performance because of system and operating limitations.

When the load becomes too big and this type of deployment is not sufficient, we need to scale out the solution and introduce a cluster of servers.

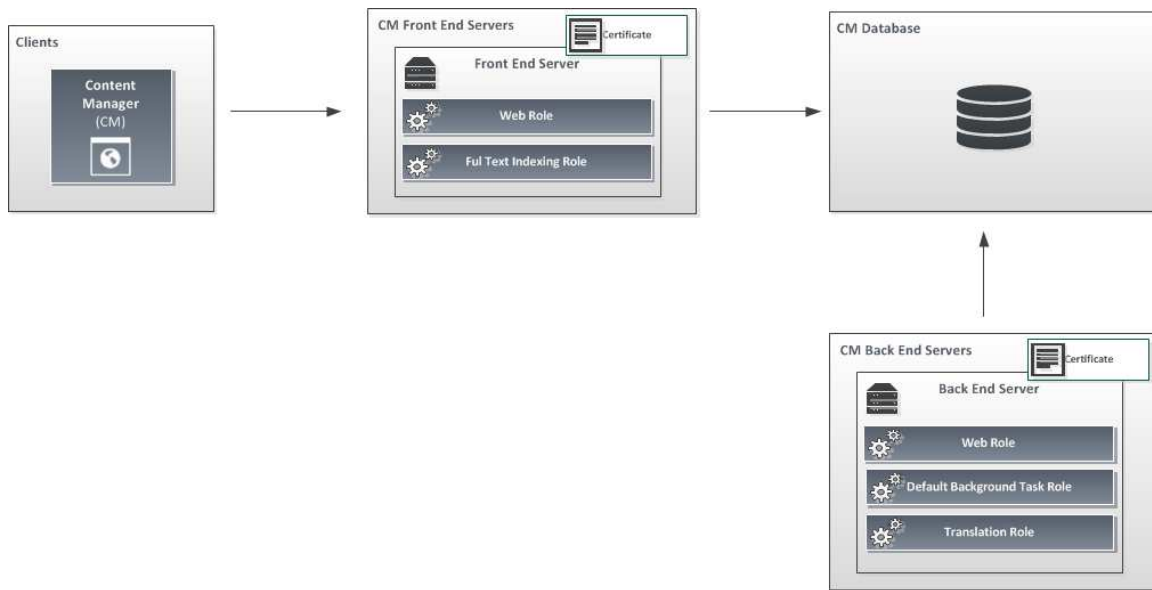
Simple cluster

The simple cluster deployment uses two servers. Each server focuses on delivering a subset of the required application functionality.

The servers types are:

- Front end server: it is responsible for all interactive and web services functionality.
- Back end server: it is responsible for all non interactive functionality.

The following diagrams shows how the two servers work together while splitting responsibilities:



Front end server

The front end server main responsibility is to provide output for web clients and web services.

Two roles are required to accomplish this:

- The Web role service allows executing web requests for the web client, and web service requests.
- The Full text indexing role provides crawling and indexing functionality for the SolrLucene search engine.

The front end server is the public facing server handling all interactive web requests. The Web role exposes public endpoints through the *Internet Information Services*, which is the default web engine on a Microsoft Windows Server operating system. All the endpoints are secured with secure sockets layer (SSL) to provide the `https` schema. Depending on the expected accessibility towards the endpoints between intranet and internet, the certificate has to be configured accordingly.

The certificate's subject name must match the hostname for the endpoints that the clients will be using. For example, if the web client is provided an endpoint like `https://front.ish.example.com/ISHCM/` (where `ish` refers to an example related to Content Manager, and `front` specifies it further as a front end server) or the web service client targets an endpoint like `https://front.ish.example.com/ISHWS/Wcf/API25/Application.svc`, then the hostname is `front.ish.example.com`. This must match the SSL certificate's subject name. Depending on the scenario, the `front.ish.example.com` should or should not match the server's fully qualified domain name.

Note: The SSL certificate's subject always matches the hostname that is visible on the client, regardless the server's fully qualified domain name (FQDN).

How to configure a front end server

Explains how to configure a server as a front end server.

Procedure

1. The Web role is enabled by default on all servers.
2. On `Internet Information Services Manager` make sure that the certificate's subject name configured on the `https` binding matches the hostname that the web clients, web service clients and client tools target.
3. Enable the Full text indexing role.

Note: We recommend deploying only one full text indexing role per database. If you scale out the front end server, we recommend having one front end server to run the Full text indexing role, and redirecting all other servers to that server.

Back end server

The back end server's main responsibility is to provide execution for all queued actions initiated by user actions or web service calls, including translation functionality.

Three roles are required to accomplish this

- The Web role allows accessing the web client and the web services from the same server.
- The Default background task role provides execution for the Background task component.
- The Translation role provides functionality for translation-related flows.

The back end server is in other words the work horse for the non-interactive and long running flows. The Web role exposes internal endpoints through the `Internet Information Services` which is the default web engine on a Microsoft Windows Server operating system. All the endpoints are secured with secure sockets layer (SSL) to provide the `https` schema. The SSL certificate has to be configured with intranet accessibility only in mind.

The certificate's subject name must match the hostname for the endpoints that the clients will be using. Since the accessibility to this server is only through intranet, the web client and web service client use fully qualified domain name (FQDN) based endpoints like `https://back.ish.example.com/ISHCM/` (where `ish` refers to an example related to Content Manager, and `back` specifies it further as a back end server) and `https://back.ish.example.com/ISHWS/Wcf/API25/Application.svc`.

The Default background task role is configured out of the box to execute all possible handlers. This means that all dependencies must be installed and properly configured on this server. For example third party renderer licenses must be properly configured when this server will be used to publish. If the dependencies are not present, the Default background task role will execute the pending items but error's will be raised.

The Translation role also requires configuration, like the target endpoints of SDL WorldServer or SDL TMS for example.

We recommend re-purposing the existing Full text indexing role, typically installed on a front end server, by forwarding the requests of the back end server.

How to configure a back end server

Explains how to configure a server as a back end server.

Procedure

1. By default, the Web role is enabled on all servers.
2. On Internet Information Services Manager, make sure that the certificate subject name in the `https` binding matches the host name the web clients and web service clients target.
3. Enable the Default background task role.
4. Enable the Translation role.
5. Redirect the full text indexing of the back end server to a front end server with the `full text indexing` role.

Redirecting full text indexing

Redirect full text indexing when configuring back end server, or when the local machine does not have full text indexing (FTI). This can be done on any server if FTI is handled on a different server, for back end servers when configuring asynchronous load balancing, or when configuring network load balancing.

About this task

The configuration for full text indexing (FTI) is handled by the Crawler and SolrLucene entries in the registry.

A default configuration is installed that connects to the localhost. For example:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Trisoft\TriDK\TriDKApp\InfoShareAu-  
thor]  
"CrawlerIndexEngineType"="ApacheSolrLucene"  
"SolrLuceneBaseUrl"="http://localhost/solr"  
"SolrLuceneHTTPTimeout"="180"  
"SolrLuceneHTTPRetries"="5"
```

Note: The localhost typically has the address 127.0.0.1 port 8080.

To redirect full text indexing on any server to the server holding the full text indexing role:

Procedure

1. On the server, alter `SolrLuceneBaseUrl` to the URL for SolrLucene on the full text indexing (FTI) server.

Example: For example, where `fti.example.com` is the FTI server's URL for SolrLucene, modify the server's registry to read:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Trisoft\TriDK\TridkApp\InfoShareAuthor] "SolrLuceneBaseUrl"="http://fti.example.com/solr"
```

Full text indexing is now directed to the server as specified in the registry for `SolrLuceneBaseUrl`.

2. Reboot the server or, at minimum, a recycle is required for the application pool and COM+ server application.
3. Disable the crawler service on the server that has been redirected to the FTI server:
 - Click **Control Panel > Administrative Tools > Services**
 - Double-click on the service named, **Trisoft InfoShare Crawler One**.
There may be more than one service. If so, follow the procedure for each.
 - Under the **General** tab, set the **Startup type** to **Disabled**.
4. To allow incoming requests on the server holding the full text index role from other servers, add all the servers IPs, or server IP range to the FTI server, in the `\App\Utilities\SolrLucene\Jetty\etc\jetty-ipaccess.xml` file's white list.

Example: For example, to allow incoming requests on the FTI server from other servers as identified by the IP address 127.0.0.1 and IP range 10.98.0-255.0-255:

Example: On the FTI server modify `jetty-ipaccess.xml` to read:

```
<Set name="white">
  <Array type="String">
    <Item>127.0.0.1</Item>
    <Item>10.98.0-255.0-255</Item>
  </Array>
</Set>
...
```

5. Make sure that the Firewall on the machine hosting the full text index role allows incoming connections on the specified IPs and/or ports.
6. Restart the **TrisoftSolrLucene** Windows service on the server holding the full text index role.

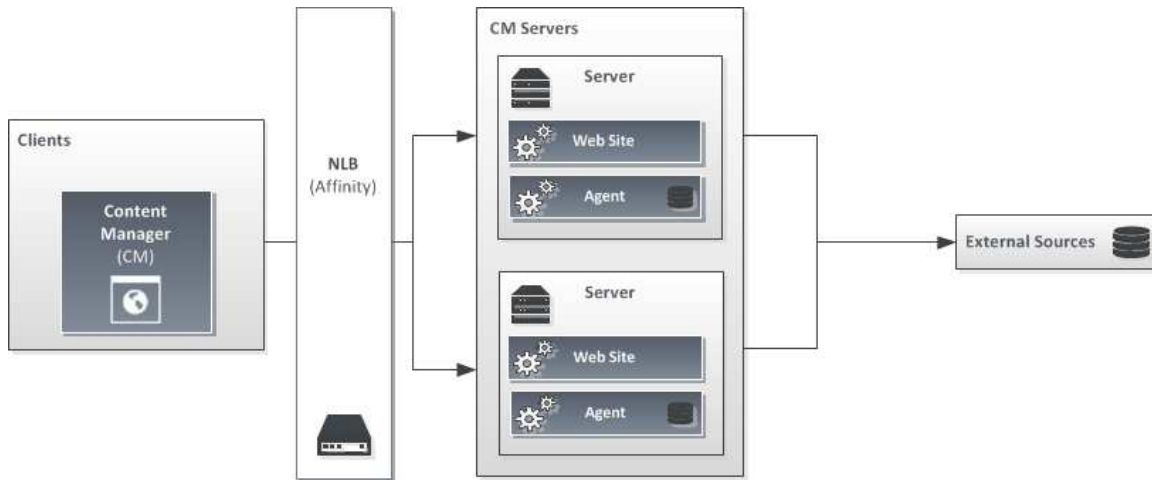
Multi server deployment

The multi server cluster deployment shows how to scale out.

All servers within the node are identical.

Creating multi server deployment and adding them as nodes on a network load balancer requires the following:

- The network load balancer has to be configured with affinity. This means that every request originating from the same client will be served always by the same server.
- The configuration between different instances of the Agent must be identical so the loaded information set is also the same across the nodes in the cluster.



Advanced server cluster

When the two server simple cluster is not sufficient to handle the load, and you want to allow faster output and bigger throughput, you can add more servers with dedicated roles.

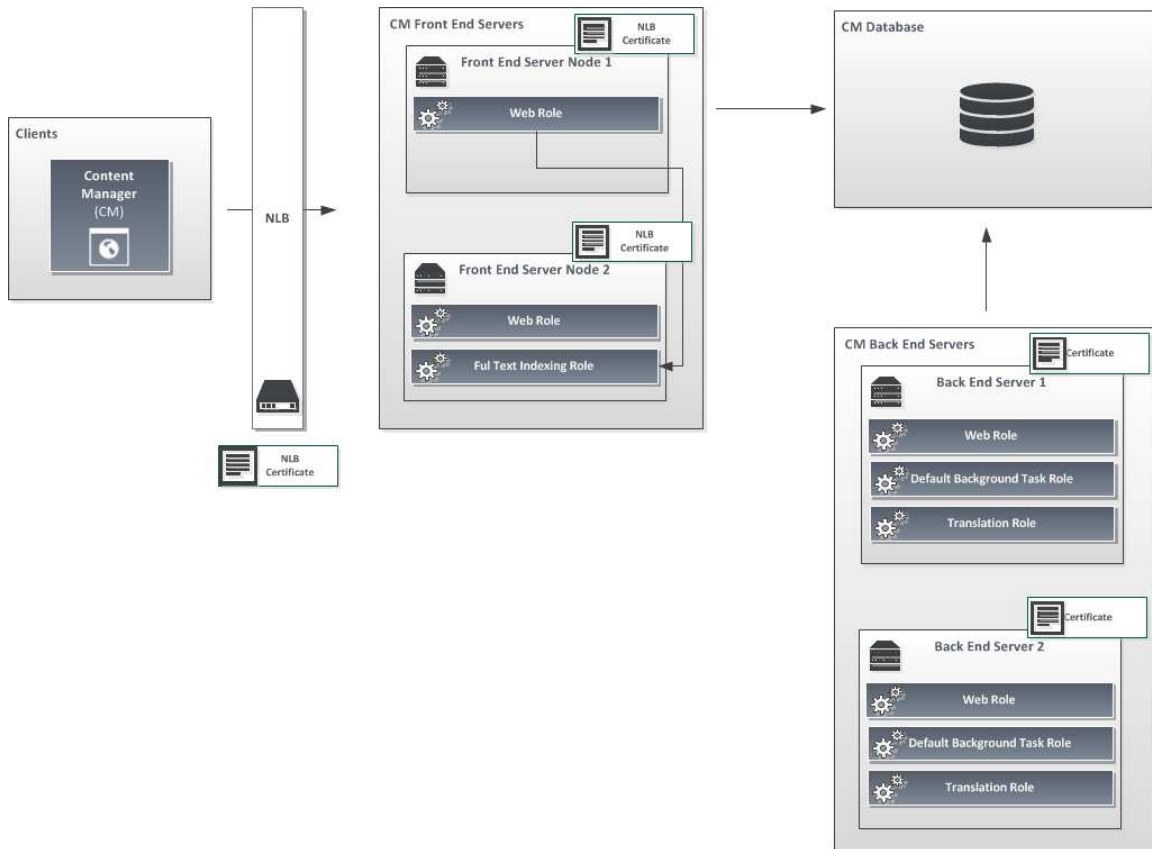
Each type of server can be scaled out. In this example case we will duplicate each server type.

So a Content Manager advanced cluster is an example of a four server deployment. With the advanced server cluster we still have the same server types as with the simple cluster, but with more aggregated power:

- The front end server is responsible for all interactive and web services functionality.
- The back end server is responsible for all non interactive functionality.

Scaling out each type of server means that we are scaling out the roles.

The following diagram shows how the two servers work together while splitting responsibilities:



To scale out a front end server, you need a network load balancer.

Note: Some roles can be scaled out by duplication even on the same server, if the hardware can handle it.

Network load balancing

Network load balancing is a technology that can be used to increase the maximum capacity and efficiency of the Web role.

A typical network load balancing deployment is a network load balancer over a cluster of identical nodes. The network load balancer is like a proxy to the nodes. The clients only see the balancer and are not aware of each of the specific nodes. This is also the main reason that the nodes have to be identical.

When a network load balancer receives a request it will forward the request to one of the nodes. There are several algorithms to drive the balancing act between the nodes. Some algorithms are restrictive e.g. affinity and some are not. Content Manager does not have any special requirements with regards to the algorithm and any node is as good as any other at the moment it joins the cluster.

Best practices to configure a node in network load balancing

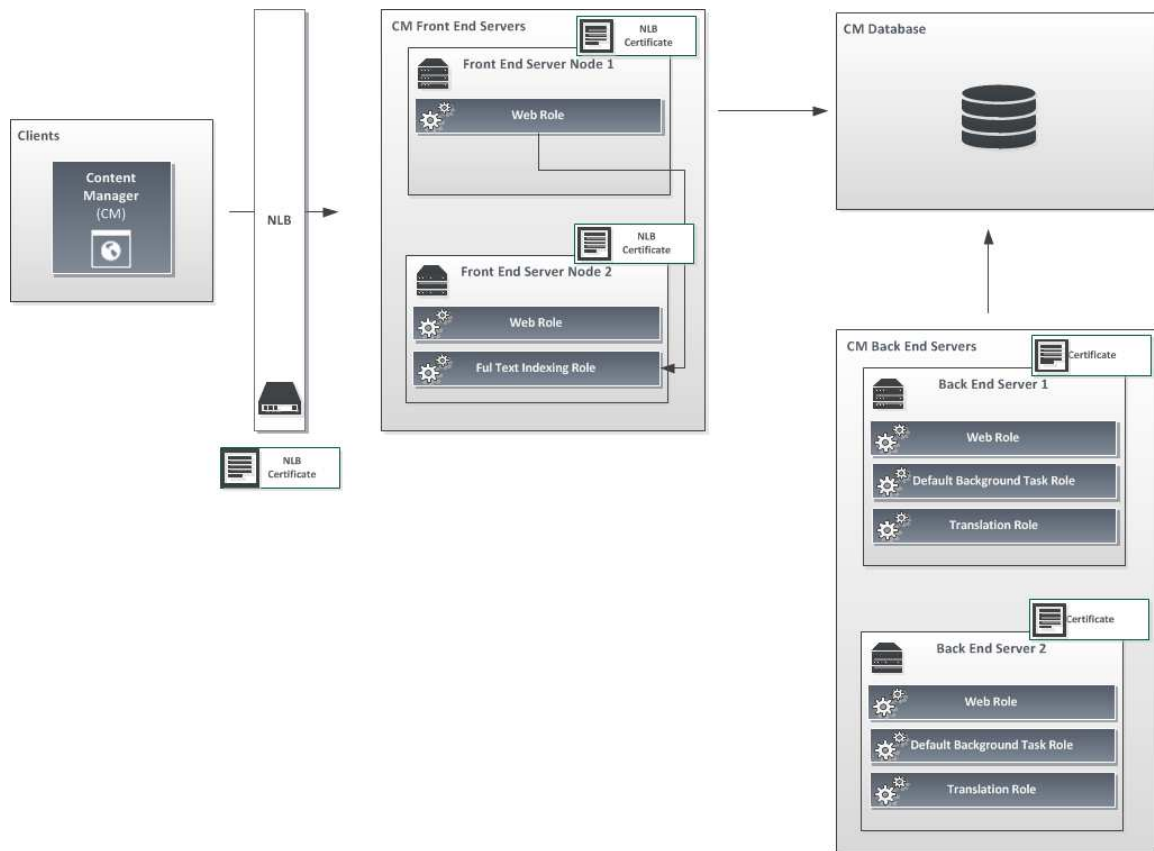
Explains how to configure a node in a network load balancing cluster.

In a typical network load balancing deployment, the network load balancer acts as a proxy to the nodes within the cluster. Because Content Manager requires traffic to be encrypted over `ssl` and `https` schema endpoints, special attention is required regarding the SSL certificate used to configure the Web role.

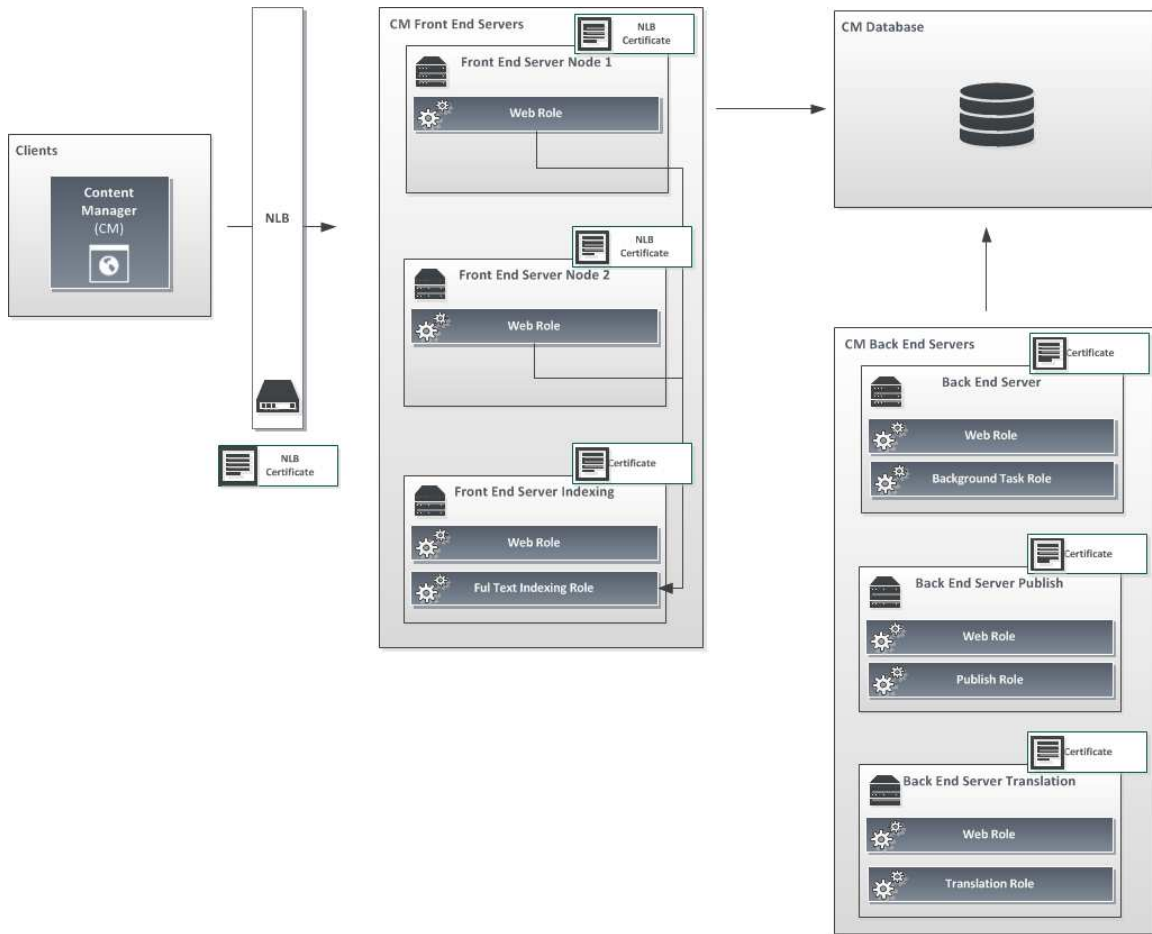
The certificate subject name must match the host name of the endpoints that the clients will be using. When a network load balancer (or any other proxy) is what the clients target, then the target hostname is e.g. `lb.ish.example.com` (where `ish` refers to an example related to Content Manager, and `lb` specifies it further as a load balancing server). For example, the web client targets an endpoint like `https://lb.ish.example.com/ISHCM/` and the web service client targets an endpoint like `https://lb.ish.example.com/ISHWS/Wcf/API25/Application.svc`. The `lb.ish.example.com` is completely independent from the server name of each node, but it forces the certificate used to configure the Web role to have this subject name. This means that a proper certificate has to be created based on the balancer's properties and shared on each Front end server before installation.

Also multiple Front end server nodes mean multiple Full text indexing roles. To force each server to deliver the same results for each search request, we need to redirect them to consume the same Full text indexing role instance in the cluster. Typically, this means that the Full text indexing role is fully active on one of the nodes and disabled on the other nodes.

In the diagram below we can identify the shared certificate used by all Front end server nodes and that only one Full text indexing role is used as the repository.



For more advanced network load balancing deployments you may even have to specialize one of the Front end server to execute only the Full text indexing role. Although the node has the Web role still active, it is not part of the cluster and it never receives requests. In this case all balanced nodes are equal with regards to throughput and are not affected by the execution of the Full text indexing role.



Best practices to specialize back end servers

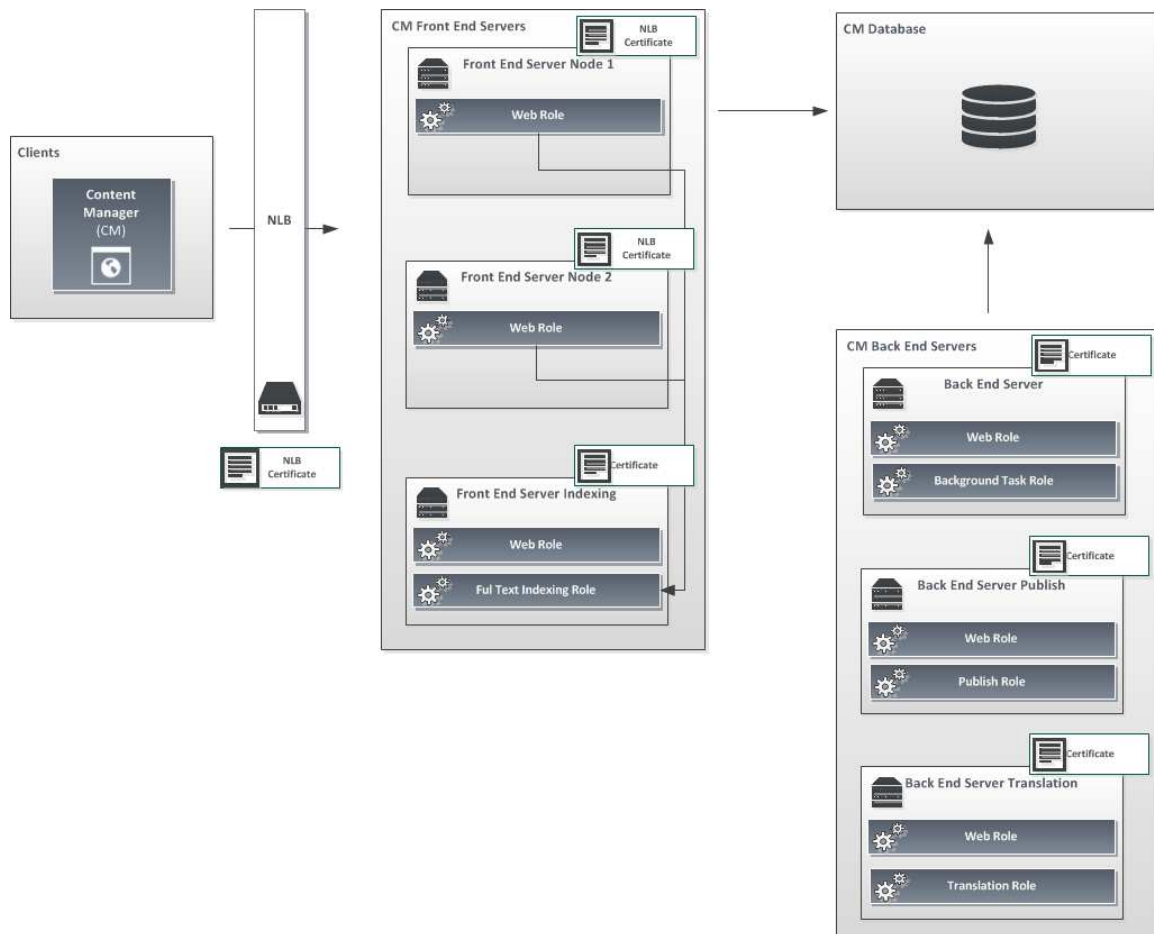
Explains how to specialize a back end server.

Unlike the Front end server configuration, back end servers in a cluster do not need to be identical.

Out of the box, a back end server is configured with the following active roles:

- Web role.
- Default background task role, where the `Default` service role configuration is targeted in the **Settings > XML Background Task Settings**.
- Translation role.

This means that all out of the box servers have the same behavior, but it is possible to differentiate. For example, we could set up a cluster of three back end servers where two servers would be specialized in the publish and translation functionality respectively and the last of the three would be configured to execute the rest. This deployment is visible in the next diagram.



However, it can be useful to create for instance three back end servers with specific roles like:

- A back end server for publishing:
 - Web role;
 - Publish role.
- A back end server specialized only in the translations:
 - Web role;
 - Translation role.

Note: Since the server is running as a dedicated translation server, the **Trisoft InfoShare BackgroundTask** service must run with a role which only contains the translation related eventTypes.

- The last back end server is required to execute all items that are not picked up by the other two. In this case all handlers relative to the publish and translation functionality will be excluded. Those handlers need to be referenced by a service role in **Settings > XML Background Task Settings**. For example:

```
<service role="Generic">
  <matrix>
```

```

<group name="SynchronizeToLiveContent" maxExecutions="1">
  <handlers>
    <add ref="SYNCHRONIZETOLIVECONTENT" />
  </handlers>
</group>
<group name="Others" maxExecutions="2">
  <handlers>
    <add ref="THUMBNAILSUBMIT" />
    <add ref="ISHBATCHIMPORT" />
  </handlers>
</group>
</matrix>
<!-- The service will check for tasks to recover (= revoke the lease)
-->
<leaseRecovery isEnabled="true" interval="00:05:00" />
<!-- If no next tasks are present, the poller will sleep the specified
period in the interval -->
<poller isEnabled="true" interval="00:00:10" />
<!-- The service will aggregate tasks with the same aggregationId,
only when the last item is submitted longer than the gracePeriod
the tasks will be aggregated.
Note: if there are no tasks to aggregate, the aggregation will
sleep the specified period in the interval -->
<aggregationRecovery isEnabled="true" gracePeriod="00:10:00" interval="
00:10:00" maximumRetries="3" />
</service>

```

Note: Although all back end servers can be configured to have enabled the recovery elements in **Settings > XML Background Task Settings**, we can also specialize to this aspect. For instance, the specialized back end servers don't do recovery and only the generic service role is enabled to recover.

Note: Further specialization is possible based on the hardware of the server. We can create in **Settings > XML Background Task Settings** service roles names like `Publish16GB` and `Publish8GB` with different value on the `maxExecutions`. The different service role names are relevant to the expected memory.

Note: Due to 32-bit process architecture restrictions, the maximum memory of each process is restricted to 2GB. To leverage a server's higher memory capacity we can setup multiple services on the same server where the service role name can be the same or different.

Server roles

An overview of the different server roles which can be recognized within a Content Manager installation

In a standard Content Manager installation some functionality relies on the combination of some components and configurations. The combination of components and configurations which provide a specific functionality are referenced as a "server role".

Web role

The web role provides all web endpoints like the web site, the web services and the internal security token service

The web role provides the following web endpoints

- ISHCM which is the Content Manager Web Client
- ISHWS which hosts all web services
- ISHSTS which is the internal Security Token Service

Since the `Web role` is required on every Content Manager installation, an out-of-the-box Content Manager installation will enable all web endpoints.

The `Web role` is used to serve external clients, but it is also used to serve the internal roles like the Translation role and the Default background task role. When it is used for external web endpoints, the `Web role` can be scaled out via Network Load Balancing (NLB).

Full text indexing role

This role groups everything for the full text search functionality by SolrLucene

The full text indexing role contains all components for the full text search functionality.

- The **Trisoft InfoShare SolrLucene** windows service that hosts and controls **SolrLucene**.
- The **Trisoft InfoShare Crawler** windows service that is responsible for gathering all data that needs to be indexed by **SolrLucene**.

How to enable the full text indexing role?

Execute the following steps in the described order to enable the full text indexing role:

- Goto **Start > Administrative Tools > Services**
- Goto the **Trisoft InfoShare SolrLucene** windows service
 - Open the **Properties**
 - Set the **Startup type** to *Automatic (Delayed Start)*
 - Click OK
 - Start the service
- Goto the **Trisoft InfoShare Crawler One** windows service
 - Open the **Properties**
 - Set the **Startup type** to *Automatic (Delayed Start)*
 - Click OK
 - Start the service
- If the server can handle more load, you can also start the **Trisoft InfoShare Crawler Two** windows service.

Remarks

We strongly advise you to allow only one deployment of this role per database. Typically this role is installed next to one of the Web roles, but it can also be a dedicated server.

Default background task role

This role provides everything which is necessary to execute all possible background tasks

The *default background task role* runs the **Trisoft InfoShare BackgroundTask** service configured with a role that contains all possible `eventTypes`.

Prerequisites

Out-of-the-box the **Trisoft InfoShare BackgroundTask One** service is configured with the `Default` role which contains all possible `eventTypes`. In this case, starting the **Trisoft InfoShare BackgroundTask One** service is enough to enable the default background task role.

However, using the following steps you can double check the configuration:

- Find the role which is used by the **Trisoft InfoShare BackgroundTask One** service
 - Goto **Start > Administrative Tools > Services**
 - Goto **Trisoft InfoShare BackgroundTask One** service
 - Click **Properties**
 - Check the value for **Path to executable**. The value should contain something like:

```
C:\InfoShare\App\BackgroundTask\Bin\BackgroundTask.exe --service
"Trisoft InfoShare BackgroundTask One" Default
```

The last parameter in the command line is the name of the service role. Out-of-the-box the role will be "Default".

- Check that the role contains all possible `eventTypes`.
 - Login to Content Manager Web Client as an administrator user.
 - Click **Settings > XML Background Task Settings**.
 - Find all `eventTypes` using `handlers/ handler/ @eventType`.
 - Goto the server definition with the role used by the service (e.g. "Default") and check that all `eventTypes` from the previous step are referenced in one of the groups. If necessary, add the missing `eventTypes`.

Note: If you had to change the configuration, you need to restart all **Trisoft InfoShare BackgroundTask** services on all servers.

Make sure that all required third-party software is installed and configured properly on this server, because if one of the dependencies is not present the background tasks will fail.

How to enable the default background task role?

Execute the following steps to enable the default background task role:

- Goto **Start > Administrative Tools > Services**
- Goto **Trisoft InfoShare BackgroundTask One** service
- Start the service

How to scale out?

There are 2 possibilities to scale out on the **Trisoft InfoShare BackgroundTask** service:

- Adding extra services with the same role
- Introducing specialized roles with a limited set of `eventTypes`.

Translation role

This role provides everything for the translation related functionality

The `Translation` role groups all components which are required for the translation related functionality:

- The **Trisoft InfoShare BackgroundTask One** service running with a role that minimally includes the following `eventTypes`:
 - `CREATETRANSLATIONFROMREPORT`
 - `CREATETRANSLATIONFROMLIST`
 - `CREATETRANSLATION`
 - `RELEASETRANSLATIONS`

These background tasks will create the necessary target language objects which can be used by the `TranslationBuilder` to be sent for translation

- The **Trisoft InfoShare TranslationBuilder One** service will group all language objects which needs to be translated for a specified translation job
- Finally, the **Trisoft InfoShare TranslationOrganizer One** service will
 - send the files to the configured translation service (SDL TMS, SDL WorldServer, ...)
 - retrieve the translated files back from the translation service (SDL TMS, SDL WorldServer, ...)
 - submit the translations back into the Content Manager repository

How to enable the translation role?

There are 2 possible scenario's for the translation role:

- Use the translation role on a dedicated translation server
- Use the translation role in combination with "Default background task role" on page 73

In the first scenario, you need to configure a new service role in **XML Background Task Settings**

1. Login to Content Manager Web Client as an administrator user
2. Click **Settings > XML Background Task Settings**.
3. Add an extra service definition with role `Translation`

```
<service role="Translation">
  <matrix>
    <group name="Translations" maxExecutions="2">
      <handlers>
        <add ref="CREATETRANSLATIONFROMREPORT" />
        <add ref="CREATETRANSLATIONFROMLIST" />
        <add ref="CREATETRANSLATION" />
        <add ref="RELEASETRANSLATIONS" />
      </handlers>
    </group>
  </matrix>
  <leaseRecovery isEnabled="true" interval="00:05:00" />
  <poller isEnabled="true" interval="00:00:10" />
  <aggregationRecovery isEnabled="true" gracePeriod="00:10:00" interval="00:10:00" maximumRetries="3" />
</service>
```

4. Adapt **Trisoft InfoShare BackgroundTask One** service to use the role `Translation`.

For both scenario's you can now continue with the following steps:

1. Configure the **TranslationBuilder** and the **TranslationOrganizer**.
2. Start all services
 - Goto **Start > Administrative Tools > Services**
 - Start the **Trisoft InfoShare TranslationBuilder One** service
 - Start the **Trisoft InfoShare TranslationOrganizer One** service
 - Start the **Trisoft InfoShare BackgroundTask One** service (if it is not running already)

Publish role

This role is the sub set of the default background task role that is responsible for exporting and publishing.

The *publish role* runs the **Trisoft InfoShare BackgroundTask** service configured with a role that contains the following `eventTypes`

- PUBLISH
- PUBLISHDITADELIVERY
- PUBLISHCONTENTDELIVERY
- INBOXEXPORT
- REPORTEXPOR
- SEARCHEXPORT
- PUBLICATIONEXPORT

Prerequisites

Make sure that all required third-party software is installed and configured properly on this server, because if one of the dependencies is not present the background tasks will fail.

How to enable the publish role?

- Configure the role in **XML Background Task Settings**:
 1. Login to Content Manager Web Client as an administrator user.
 2. Click **Settings > XML Background Task Settings**.
 3. Add an extra service definition with role `Publish`:

```
<service role="Publish">
  <matrix>
    <group name="Export" maxExecutions="2">
      <handlers>
        <add ref="INBOXEXPORT" />
        <add ref="REPORTEEXPORT" />
        <add ref="SEARCHEXPORT" />
        <add ref="PUBLICATIONEXPORT" />
        <!-- New publish -->
        <add ref="PUBLISH" />
        <add ref="PUBLISHDITADELIVERY" />
        <add ref="PUBLISHCONTENTDELIVERY" />
        <!-- Legacy publish -->
        <add ref="EXPORTFORPUBLICATION" />
      </handlers>
    </group>
  </matrix>
  <leaseRecovery isEnabled="true" interval="00:05:00" />
  <poller isEnabled="true" interval="00:00:10" />
  <aggregationRecovery isEnabled="false" gracePeriod="00:10:00"
    interval="00:10:00" maximumRetries="3" />
</service>
```

- Create a **Trisoft InfoShare BackgroundTask** service with the role `Publish`.
- Start the service.

Best practices for creating a Trisoft InfoShare BackgroundTask service with a specific role

The topic described how to create a Trisoft InfoShare BackgroundTask service with a specific role

Of course, there are multiple ways to make a **Trisoft InfoShare BackgroundTask** service run with a specific role.

However, if possible try to create the **Trisoft InfoShare BackgroundTask** service immediately with the correct role configured by adapting the install plan.

If you want to adapt the role of an existing background task service after the installation, refer to the corresponding section in the documentation.

How to create a new BackgroundTask service with a role

This topic explains how to adapt the install plan to create a Trisoft InfoShare BackgroundTask service with the specified service role.

Before you begin

- There is no Content Manager installation yet.
- Check the name of the service role via **Settings > XML Background Task Settings**. If the name is `newServiceRole`, there should be a XML fragment like the following:

```
<service role="newServiceRole">
  <matrix>
    ...
  </matrix>
  <leaseRecovery isEnabled="true" interval="00:05:00" />
  <poller isEnabled="false" interval="00:00:10" />
  <aggregationRecovery isEnabled="false" gracePeriod="00:10:00" inter-
    val="00:10:00" maximumRetries="3" />
</service>
```

About this task

The following procedure describes how to adapt the out-of-the-box configuration of the Trisoft InfoShare BackgroundTask One windows service before installing. Of course, you can also add extra services by adapting the install plan.

Procedure

1. Open the install plan (`__InstallTool\installplan.xml`) from a Content Manager CD
2. Goto to the service definition for Trisoft-InfoShare-BackgroundTask

```
<serviceapp name="Trisoft-InfoShare-BackgroundTask">
  <servicename>Trisoft InfoShare!#installtool:PROJECTSUFFIX#!# Back-
    groundTask One</servicename>
  <filepath>#!#installtool:APPPATH#!#\App#!#installtool:PROJECTSUFFIX#!#
    \BackgroundTask\Bin\BackgroundTask.exe --service "Trisoft InfoShare#!#
    installtool:PROJECTSUFFIX#!# BackgroundTask One" Default</filepath>
  ...
</serviceapp>
```

3. Replace the `Default` role with `newServiceRole`

```
<serviceapp name="Trisoft-InfoShare-BackgroundTask">
  <servicename>Trisoft InfoShare!#installtool:PROJECTSUFFIX#!# Back-
    groundTask One</servicename>
  <filepath>#!#installtool:APPPATH#!#\App#!#installtool:PROJECTSUFFIX#!#
    \BackgroundTask\Bin\BackgroundTask.exe --service "Trisoft InfoShare#!#
    installtool:PROJECTSUFFIX#!# BackgroundTask One" newServiceRole</
    filepath>
  ...
</serviceapp>
```

4. Save the modified install plan

Results

An install plan that will install a Trisoft InfoShare BackgroundTask One windows service with the specified service role

How to adapt the role of an existing BackgroundTask service

This topic explains how to adapt the role of an existing Trisoft InfoShare BackgroundTask service.

Before you begin

Check the name of the service role via **Settings > XML Background Task Settings**.

If the name is `newServiceRole`, there should be a XML fragment like the following:

```
<service role="newServiceRole">
  <matrix>
    ...
  </matrix>
  <leaseRecovery isEnabled="true" interval="00:05:00" />
  <poller isEnabled="false" interval="00:00:10" />
  <aggregationRecovery isEnabled="false" gracePeriod="00:10:00" interval="00:10:00" maximumRetries="3" />
</service>
```

Warning: Using Registry Editor incorrectly can cause serious problems that may require you to reinstall your operating system.

Procedure

1. Modify the `Default` service role configured for the Trisoft InfoShare BackgroundTask One windows service.
 - a. Open the Registry Editor with Administrator rights.
 - b. Open the key `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Trisoft InfoShare BackgroundTask One`.
 - c. Open the value with name `ImagePath`.

The current data looks like.

```
D:\InfoShare\App\BackgroundTask\Bin\BackgroundTask.exe --service
"Trisoft InfoShare BackgroundTask One" Default
```

where `Default` is the default service role name configured out-of-the-box and present in the XML Background Task Settings.

- d. Edit the data and change `Default` to the new service role name `newServiceRole`

The data should now look like.

```
D:\InfoShare\App\BackgroundTask\Bin\BackgroundTask.exe --service
"Trisoft InfoShare BackgroundTask One" newServiceRole
```

- e. Save the data.

At this point we have configured the Trisoft InfoShare BackgroundTask One windows service to run with the `newServiceRole`.

2. Start the Trisoft InfoShare BackgroundTask One windows service

Firewalls and blocked ports

There is a variety of possibilities regarding network and firewall configurations. Only some of typical firewall configurations are described. A user knowledgeable about networking can infer the required ports and protocol settings needed for more complex configurations.

Note: The following description is intended to guide you in your network and firewall configuration. Its intention is not to be a complete how-to guide for setting up firewalls. Several settings are subject to change in newer software versions. Be certain to refer to the latest reference materials.

Network configuration using a single firewall

The first firewall is located between the Internet and the internal network

This is the first line of protection from the world wide web. All information passed through the Content Manager web sites or web services are based on the HTTPS protocol.

The techniques described in the section for HTTPS (SSL) could be required depending on the task of the Content Manager server.

Network configuration using two firewalls

The first firewall is located between the Internet and the DMZ as described above and the second is located between the DMZ and the intranet

The second line of protection protects servers which are open to the general public from the more critical company intranet servers. The zone between the first and second line of protection is also called the DMZ (DeMilitarized Zone).

The following techniques could be required depending on the task of the Content Manager server

- SMTP - when SMTP communication is required
- Database engines
 - Microsoft SQL Server access is required
 - Oracle RDBMs access is required
- HTTP(S)

SMTP

Ports and protocols used by the Simple Mail Transfer Protocol (SMTP).

Simple Mail Transfer Protocol (SMTP) is the standard for e-mail transmissions across the internet. Formally SMTP is defined in RFC 821 (STD 10) as amended by RFC 1123 (STD 3) chapter 5. The protocol used today is also known as ESMTP, and it is defined in RFC 2821.

Used by:

- Content Manager Publishing components
- Content Manager Author components

Settings:

What	Protocol	Port	Direction
SMTP	TCP	25	IN/OUT

For more information, refer to:

- <http://msdn2.microsoft.com/en-us/library/ms942998.aspx>
- <http://en.wikipedia.org/wiki/SMTP>

Microsoft SQL Server ports

Ports and protocols used by Microsoft SQL Server database engine (SQLServer).

About this task

Microsoft SQL Server is a relational database management system (RDBMS) produced by Microsoft. Its primary query language is Transact-SQL, an implementation of the ANSI/ISO standard Structured Query Language (SQL) which is used by Microsoft. You need to allow distant users to connect to the SQL server so they can address it their queries.

SQL Server is used by:

- Content Manager End User components;
- Content Manager Author components;
- Content Manager Database.

Note: The first step of this procedure is sufficient in most cases. Go through the other steps if you encounter any issue.

Procedure

1. Enable remote connections to your SQL Server.
 - a. Open **SQL Server Management Studio**.
 - b. Right-click your server's name and select **Properties**.
 - c. Tick the checkbox **Allow remote connections to this server**.
 - d. Select **OK**.

Microsoft SQL Server by default uses TCP 1433 but this can be changed using **SQL Server Enterprise Manager** or the database **Management Studio**.

2. Enable TCP/IP.
 - a. Open the **SQL Server Configuration Manager**.
 - b. In **SQL Server Network Configuration** select **Protocols for [yourServerInstance]**.
 - c. In the right-hand pane, make sure that **TCP/IP** is **Enabled**.
3. Open the 1433 port in your firewall.
 - a. In the **SQL Server Configuration Manager**, right-click **TCP/IP** and select **Properties**.
 - b. Select the **IP Addresses** tab and make sure the **TCP Port for IP1** is **1433**.
4. If you are using a named instance, create an extra rule in your firewall with the port 1434.

Note: For a named SQL Server instance (e.g. [yourServerInstance] \ SQL2012SP2), the firewall needs an extra rule on the UDP protocol with the specific port 1434. Without this rule the system will return the exception error: 26 - Error Locating Server/Instance Specified.

- a. Display the firewall advanced settings by navigating to **Control Panel > System and Security > Windows Firewall > Advanced settings**.
- b. Select **Inbound Rules** in the left-hand pane, then click **New Rule** in the right-hand pane.
- c. In the **New Inbound Rule Wizard, Rule Type step**, select **Port**.
- d. **Protocols and Ports** step, select **UDP** and set **Specific local ports** to 1434.
- e. **Action** step, select **Allow the connection**.
- f. **Profile** step, tick the **Domain** checkbox.
- g. **Name** step, enter a name for this rule, e.g. Named instance port 1434.
- h. Select **Finish**.

What to do next

For more information, refer to:

- <http://support.microsoft.com/kb/287932>
- <http://msdn.microsoft.com/en-us/library/ms942998.aspx>

Oracle RDBMS

Ports and protocols used by the Oracle RDBMS.

Oracle Database, Oracle RDBMS, or simply Oracle is a relational database management system (RDBMS) software product released by Oracle Corporation that has become a major feature of database computing.

Used by:

- Content Manager End User components
- Content Manager Author components
- Content Manager Database

The Oracle database server communicates with the Content Manager application server using a port, so make sure this port is opened IN/OUT in your firewall. The port by default is 1521.

What	Protocol	Port	Direction
Listener (runs on the database server)	TCP	1521 (default)	IN/OUT

HTTPS (SSL)

Ports and protocols used by Microsoft Internet Information Server (IIS).

HTTPS is a URI scheme used to indicate a secure HTTP connection. It is syntactically identical to the `http://` scheme normally used for accessing resources using HTTP. The `https:` URL indicates that HTTP is to be used but with a different default TCP port (443) and an additional encryption/authentication layer between the HTTP and TCP.

HTTPS is not a separate protocol, but refers to the combination of a normal HTTP interaction over an encrypted Secure Sockets Layer (SSL). An `https:` URL may specify a TCP port. If it does not, the connection uses port 443.

Used by:

- Content Manager End User Website
- Content Manager Author Website
- Content Manager WebServices

Settings:

What	Protocol	Port	Direction
HTTPS	TCP	443	IN/OUT

For more information, refer to:

- <http://msdn2.microsoft.com/en-us/library/ms942998.aspx>
- <http://en.wikipedia.org/wiki/Https>

Background task component

Background task is an application logic that is triggered on certain events, does not require user interaction and runs in a background by a background task service. Typical example is publishing process: it has to be triggered by user, but after it is triggered, it does not require user input neither does it require user to wait. Instead, the publishing process runs in a background, and user can know that it is finished by periodically checking the status of a publication.

Starting background tasks

Typically tasks are created by plugins. For example, it is possible to register a plugin that will run when user changes the status of the topic and create the background task. Typically, background task does not start executing immediately after it is created. Instead it is added to the queue from where it can be later picked up by a background task service which executes the task. This allows better distribution of load since the task can be picked up by the service (or server, because background task services can run on different servers) which is less busy.

Executing background tasks

There is only one background tasks queue which is available to every background task service. Under the hood, it is implemented as a database table, which means that once added, task will not get lost. Practically it means that task will survive the server reboot, and even if the task execution fails in the middle, task remains in the queue and can be re-started.

Every background task has an event type it is created for. For example, when you trigger the publish, there will be a background task created with event type `EXPORTFORPUBLICATION`. Task can be picked up by one of installed background task services. You can limit event types that service is allowed to pick up and amount of background tasks with the same event type that can be executed in parallel.

When task fails for one reason or another, it is automatically re-tried later. You can adjust this behavior by changing the limit of retry attempts.

Monitoring background tasks

Typically, background tasks update the status of the execution by writing to the event log. For example, background task that executes publishing will update the corresponding Publish event.



There is only one queue for background tasks. There can be any number of background task services picking up tasks.

Overview of the background task configuration

The background task configuration contains all information for running the background task services and handling the background tasks.

Introduction

The configuration is stored inside the Content Manager database and is accessible via the Content Manager web client using **Settings > XML Background Task Settings**.

Tip: For detailed information, check the `Admin.XMLBackgroundTaskConfiguration.xml` file.

Configuration for the background task services

The configuration can contain different types of services. Every background service runs with a specific *role*. For every role the configuration describes the behavior:

- Should the service execute background tasks?
- How often should it poll for a new background task?
- Should the service recover failed background tasks?

If the service is configured to execute background tasks, the configuration specifies for which groups of event types the service is responsible. The configuration defines for every group how many background tasks are allowed to run in parallel.

Out-of-the-box, the services are installed with the 'Default' role and will pick up all possible background tasks. However, it is possible to configure for instance a service with the role 'Publish' picking up only the background tasks with event type `EXPORTFORPUBLICATION`

Configuration for the handlers

The configuration contains a list with handlers. Each handler is handling one *eventType* and can be executed synchronously or asynchronously. The handler is responsible for (1) starting the activator which will execute the background task and (2) handling any exception which occurs.

Per type of activator, the configuration not only contains the necessary information to create and run the activator, but it also contains the configuration with parameters that are used during the execution of the background task.

Note: These parameters can contain (environment) variables that are resolved by the background task service configuration.

The configuration also specifies if the background task must be executed within the same process (of the service) or within a new process, and for how long the background task is allowed to execute. Each background task is executed within the security context of the user that submitted the background task. How the security context is created depends on the authorization type.

When the execution of the background fails, the configuration indicates for each error number if the background task must be retried and how many times the background task can be scheduled to re-execute.

Usage of variables inside the background task configuration

Referencing environment variables inside the background task configuration is useful when exact configuration value varies from server to server and therefore cannot be single-sourced. Environment variables are resolved at the moment background task service is being initialized with the actual values set on that specific server.

Normally background task gets its parameters from the background task configuration. This way the background task parameters are defined in a single place and can be easily accessed by the task regardless of which server or service executes it.

However, sometimes it is not easy (or even possible) to provide a value that would work on every server. The typical case is the file path, which may differ from one server to another. For example, PUBLICATION-EXPORT event type needs to know the export location, which can be a different folder depending which server picks up the task.

To solve this problem, background task configuration allows referencing environment variables. Environment variable can be provided as a value of any element or attribute.

PUBLICATIONEXPORT references %ISHPROJECTDATAPATH% in the value of *exportlocation* and *exportspeclocation* parameters.

```
<handler eventType="PUBLICATIONEXPORT">
  <scheduler executeSynchronously="false" />
  <authorization type="authenticationContext" />
  <execution timeout="01:00:00" recoveryGracePeriod="00:10:00"
    isolationLevel="Process" useSingleThreadApartment="true" />
  <activator>
    <comIEventHandler projectName="IshPluginsIso" className="cout">
      <configuration>
        <parameters>
          <parameter name="exportlocation" type="value">%ISHPROJECTDATAPATH%
            \ExportService\Data\DataExports</parameter>
```

```

<parameter name="exportspeclocation" type="value">%ISHPROJECTDATAPATH%
\ExportService\Data\WatchFolder</parameter>
<parameter name="separateIng" type="value">yes</parameter>
<parameter name="requestedmetadata" type="ishfields">
<ishfields>
<ishfield name="FSTATUS" level="lng" />
</ishfields>
</parameter>
<parameter name="raiseevent" type="value">ZIPFILES</parameter>
<parameter name="filenameprefix" type="ishfields">
<ishfields>
<ishfield name="FTITLE" level="logical" />
</ishfields>
</parameter>
</parameters>
</configuration>
</comIEventHandler>
</activator>
<errorHandler maximumRetries="0" />
</handler>

```

When the background task service is being initialized, every environment variable is replaced with the actual value.

Remember: It is your responsibility to make sure that every environment variable referenced in the background task configuration is set!

Remember: Setting environment variable to empty string deletes the environment variable!

The easy way to set the environment variables for the lifetime of the background task service is to add them to the background task service configuration file. The file is located on the Content Manager server: `\InfoShare\App\BackgroundTask\Bin\BackgroundTask.exe.config`

Variables are configured in the **variables** element within the section **trisoft.infoShare**.

backgroundTask. Background task service will read these values during initialization and use them to set the actual environment variables.

Providing the environment variable values for PUBLICATIONEXPORT in the background task service configuration file.

```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
<configSections>
<section name="trisoft.infoShare.backgroundTask" type="Trisoft.
InfoShare.BackgroundTask.BackgroundTaskConfigurationSection, Trisoft.
InfoShare.BackgroundTask, Version=11.0.0.0, Culture=neutral,
PublicKeyToken=555d9fcb450e0935"/>
<!-- Other <section> and <sectionGroup> elements. -->
</configSections>
<startup>
<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
</startup>
<trisoft.infoShare.backgroundTask>
<variables>
<!-- Value cannot be an empty string! -->
<add key="ISHPROJECTAPPPATH" value="C:\InfoShare\App" />

```

```
<add key="ISHPROJECTDATAPATH" value="C:\InfoShare\Data" />
</variables>
</trisoft.infoShare.backgroundTask>
</configuration>
```

Understanding the availability matrix

The availability matrix defines which event types and how many of their instances is the background task service allowed to execute in parallel.

Executing background tasks consumes system resources, most importantly CPU time and system memory. Deciding which kind of tasks and how many instances of them are allowed to run in parallel on the same server is important because this will affect stability, throughput and overall system performance. This decision is a compromise: increasing the task parallelism can increase the throughput (more tasks are executed within the same amount of time), but it puts the system under pressure and may even result in situations when your process on the server runs out of memory.

The availability matrix is configured in the background task configuration, per server role, under matrix element. The default background task configuration is located in `\Websites\Author\EnterViaUI\Admin.XMLBackgroundTaskConfiguration.xml`, which is delivered on every CD, and contains the latest suggested out of box values for the matrix.

```
<matrix>
  <group name="Translations" maxExecutions="2">
    <handlers>
      <add ref="CREATETRANSLATIONFROMREPORT" />
      <add ref="CREATETRANSLATIONFROMLIST" />
      <add ref="CREATETRANSLATION" />
      <add ref="RELEASETRANSLATIONS" />
    </handlers>
  </group>
  <group name="Export" maxExecutions="2">
    <handlers>
      <add ref="EXPORTFORPUBLICATION" />
      <add ref="INBOXEXPORT" />
      <add ref="REPORTEXPORT" />
      <add ref="SEARCHEXPORT" />
      <add ref="PUBLICATIONEXPORT" />
    </handlers>
  </group>
  <group name="SynchronizeToLiveContent" maxExecutions="1">
    <handlers>
      <add ref="SYNCHRONIZETOLIVECONTENT" />
    </handlers>
  </group>
  <group name="Others" maxExecutions="2">
    <handlers>
      <add ref="THUMBNAILSUBMIT" />
      <add ref="ISHBATCHIMPORT" />
    </handlers>
  </group>
</matrix>
```

Matrix defines groups of handlers with common features or requirements or functionality. For every group, `maxExecutions` attribute specifies how many instances of the specified background task handler can be executed concurrently. The availability matrix works proactively by controlling what the background task service will poll each time. Only tasks that are valid for the current state of execution and the availability matrix are allowed to begin executing.

Here is an example flow to better understand what happens when the service begins to execute.

Looking at the first group Translations, the service will try to execute any background task with the configured handler for example `CREATETRANSLATIONFROMREPORT`. While a `CREATETRANSLATIONFROMREPORT` instance is executing the service is allowed to pick one more item from the queue matching the configured handlers of this specific group including `CREATETRANSLATIONFROMREPORT`, for example `RELEASETRANSLATIONS`. As long as both background tasks are executing the service is not allowed to execute any more from this group because the limit 2, defined in `maxExecutions` attribute, is reached. Once one of the tasks finishes then the service is allowed once again to execute background tasks from the this group. The above reasoning is applied to every configured group and the service will always try to execute a background task from any group that still hasn't reached its `maxExecutions` limit.

Important: You can only put one event type in one group. If you configure the same event type in 2 different groups, you will get an error when trying to submit this configuration.

Default background task configuration comes with 2 background task service roles. The Console background task service role is optimized for testing through the console mode. This role will restrict the process to execute a maximum of 1 background task at any given time.

Understanding the isolation level of the handler

Explains the significance of isolation level configuration value for a background task handler.

Every handler configuration in the Background task XML settings defines an `isolationLevel` within the `execution` element.

The isolation level is allowed to have one of the following values

- None
- Process

When the value is `None` then all background tasks of this handler will execute within the process of the background task service.

Every operating system process has a limited amount of resources that it can access. With the background task service, we are interested in the memory limitation. The process's maximum memory must be shared between the requirements of the service's components but also the running background tasks.

There is also the potential of memory leaks than can be caused by a background task. Although the background task service is optimized against memory leaks it can run out of memory because a background task had misused the memory.

Different combinations can result to an unstable background task service process or handler that runs out of memory. To protect the background task service but also provide an isolated memory space to a specific background task, the `Process` value was introduced for the `isolationLevel`. When this configuration is enabled for a handler, the background task service will spawn a new process with the sole goal to execute this specific background task instance. This way the execution is isolated within the memory space of a specific process that is dedicated fully for the background task. Also any memory leak caused by the handler's execution is limited to the lifetime of this process and has no effect to the background task service process. There is an overhead though. A new process means that everything has to be loaded resulting to slower startup times of the actual execution. The total overhead depends on the load on the server.

This way the handler receives maximum memory space and also the background task service is protected against memory leaks. The only tradeoff is a potential overhead in the total execution time.

Based on the above, here are some suggestions to help you get to the correct value for `isolationLevel` of a handler:

- If the handler of the background task requires a lot of memory then it must be isolated.
- If the handler has the potential to execute for long then it should be isolated.
- Choosing `Process` for the `isolationLevel` should take into account the overhead of the startup time compared to the average actual execution time of the handler

Out of the box configuration has all handlers configured to execute with `isolationLevel` set to `None`. Only for `EXPORTFORPUBLICATION` is configured to execute with `isolationLevel` set to `Process` because it is very memory intensive. Because it has the potential to execute for long the extra overhead in startup time is small relative to the average expected execution time.

Installing desktop client tools

You can choose to install desktop client tools (Authoring Bridge, Publication Manager, Condition Manager and Content Importer) based on the role and responsibilities of the user. The versions of the desktop clients tools must match the version of Content Manager installed on the server.

Installing the Authoring Bridge

The Authoring Bridge is used to access the repository using an authoring tool.

About this task

Choose the Authoring Bridge installer for the XML editor software that you are using (XMetaL, oXygen, ...). The type and version of the editor you are using must be qualified with the installed version of Content Manager.

If an incompatible version of the .NET runtime framework is installed, the installer will install a correct one after your validation.

The install package's names match the following format, one for each compatible XML Editor:

- `<date>.AuthoringBridge.<version>-XMetaL-Connector.msi`
- `<date>.AuthoringBridge.<version>-oXygen-Connector.msi`
- `<date>.AuthoringBridge.<version>-ArbortextEditor-Connector.msi`

The development kit's executable name matches the following format:

- `<date>.AuthoringBridgeSDK.<version>.exe`

Procedure

1. Double-click the Authoring Bridge install package, or **Setup Wizard**, and follow the proposed list of steps. After you hit the **Finish** button, the installation is complete and the SDL Tridion Docs menu is now available in your authoring tool's menu bar.

Results

When you first try to access the Repository from your authoring tool, you will be requested to specify a user account in order to establish the connection.

For uninstall, be aware of the following:

- If you are using XMetaL on Windows, you must uninstall the Authoring Bridge before uninstalling XMetaL.
- If the Authoring Bridge is uninstalled, the SDL Tridion Docs menu will remain in the menu bar, with all menu options greyed out. You must start XMetaL while pressing the **CTRL** key in order to have XMetaL clean up the menu bar and remove the SDL Tridion Docs menu.

Installing Publication Manager

The publication manager is used to create, modify, check the status of, and produce publications in various formats.

About this task

If an incompatible version of the .NET runtime framework is installed, the installer will install a correct one after your validation.

The install package's name matches the following format: <date>.PublicationManager.<version>.msi

Procedure

1. Double-click the Publication Manager install package, or **Setup Wizard**, and follow the proposed list of steps. After you hit the **Finish** button, the installation is complete and Publication Manager is now available for you to use.

Installing Condition Manager

The Condition Manager is used to create and manage conditions for your publications.

About this task

If an incompatible version of the .NET runtime framework is installed, the installer will install a correct one after your validation.

The install package's name matches the following format: <date>.ConditionManager.<version>.msi

Procedure

1. Double-click the Condition Manager install package, or **Setup Wizard**, and follow the proposed list of steps.
2. Select the **Enable Synchronization** option when proposed (on the tab where you specify the installation folder). Content Manager can synchronize the set of conditions with definitions from another system. Synchronization needs to be configured separately, **Enable Synchronization** only makes the menu items available in the application.
3. After you hit the **Finish** button, the installation is complete and Condition Manager is now available for you to use.

Installing Content Importer

Content Importer helps you import content through the Content Manager Web Client commands.

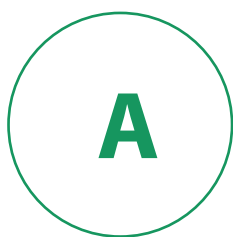
About this task

If an incompatible version of the .NET runtime framework is installed, the installer will install a correct one after your validation.

The install package's name matches the following format: `<date>.ContentImporter.<version>.msi`

Procedure

1. Double-click the Publication Manager install package, or **Setup Wizard**, and follow the proposed list of steps. After you hit the **Finish** button, the installation is complete and Content Importer had added functionality to the Content Manager Web Client.



Acknowledgments

SDL products include open source or similar third-party software.

7zip

Is a file archiver with a high compression ratio. 7-zip is delivered under the GNU LGPL License.

7zip SFX Modified Module

The SFX Modified Module is a plugin for creating self-extracting archives. It is compatible with three compression methods (LZMA, Deflate, PPMd) and provides an extended list of options. Reference website <http://7zsfx.info/>.

Akka

Akka is a toolkit and runtime for building highly concurrent, distributed, and fault tolerant event-driven applications on the JVM.

Amazon Ion Java

Amazon Ion Java is a Java streaming parser/serializer for Ion. It is the reference implementation of the Ion data notation for the Java Platform Standard Edition 8 and above.

Amazon SQS Java Messaging Library

This Amazon SQS Java Messaging Library holds the Java Message Service compatible classes, that are used for communicating with Amazon Simple Queue Service.

ANTLR

ANTLR is a powerful parser generator that you can use to read, process, execute, or translate structured text or binary files.

Apache ActiveMQ

Apache ActiveMQ is the most popular and powerful open source messaging and Integration Patterns server.

Apache Ant

Apache Ant is a Java library and command-line tool whose mission is to drive processes described in build files as targets and extension points dependent upon each other. The main known usage of Ant is the build of Java applications. Ant supplies a number of built-in tasks allowing to compile, assemble, test and run Java applications. Ant can also be used effectively to build non Java applications, for instance C or C++ applications. More generally, Ant can be used to pilot any type of process which can be described in terms of targets and tasks.

Apache Commons BeanUtils

The Java language provides *Reflection* and *Introspection* APIs (see the `java.lang.reflect` and `java.beans` packages in the JDK Javadocs). However, these APIs can be quite complex to understand and utilize. The *BeanUtils* component provides easy-to-use wrappers around these capabilities.

Apache Commons Codec

Apache Commons Codec (TM) software provides implementations of common encoders and decoders such as Base64, Hex, Phonetic and URLs.

Apache Commons Configuration

The Commons Configuration software library provides a generic configuration interface which enables a Java application to read configuration data from a variety of sources. Commons Configuration provides typed access to single, and multi-valued configuration parameters.

Apache Commons DBCP (Database Connection Pools)

Many Apache projects support interaction with a relational database. Creating a new connection for each user can be time consuming (often requiring multiple seconds of clock time), in order to perform a database transaction that might take milliseconds. Opening a connection per user can be unfeasible in a publicly-hosted Internet application where the number of simultaneous users can be very

large. Accordingly, developers often wish to share a "pool" of open connections between all of the application's current users. The number of users actually performing a request at any given time is usually a very small percentage of the total number of active users, and during request processing is the only time that a database connection is required. The application itself logs into the DBMS, and handles any user account issues internally. There are several Database Connection Pools already available, both within Apache products and elsewhere. This Commons package provides an opportunity to coordinate the efforts required to create and maintain an efficient, feature-rich package under the ASF license.

Apache Commons FileUpload

The Commons **FileUpload** package makes it easy to add robust, high-performance, file upload capability to your servlets and web applications.

Apache Commons HttpClient

HttpClient was started in 2001 as a subproject of the Jakarta Commons, based on code developed by the Jakarta Slide project.

Apache Commons Lang

The standard Java libraries fail to provide enough methods for manipulation of its core classes. Apache Commons Lang provides these extra methods.

Lang provides a host of helper utilities for the java.lang API, notably String manipulation methods, basic numerical methods, object reflection, concurrency, creation and serialization and System properties. Additionally it contains basic enhancements to java.util.Date and a series of utilities dedicated to help with building methods, such as hashCode, toString and equals.

Apache Commons Logging

The Logging package is an ultra-thin bridge between different logging implementations. A library that uses the commons-logging API can be used with any logging implementation at runtime. Commons-logging comes with support for a number of popular logging implementations, and writing adapters for others is a reasonably simple task.

Apache Commons Pool

Pool provides an Object-pooling API, with three major aspects:

1. A generic object pool interface that clients and implementers can use to provide easily interchangeable pooling implementations.
2. A toolkit for creating modular object pools.
3. Several general purpose pool implementations.

Apache FOP

Apache FOP (Formatting Objects Processor) is a print formatter driven by XSL formatting objects (XSL-FO) and an output independent formatter. It is a Java application that reads a formatting object (FO) tree and renders the resulting pages to a specified output. Output formats currently supported include PDF, PS, PCL, AFP, XML (area tree representation), Print, AWT and PNG, and to a lesser extent, RTF and TXT. The primary output target is PDF.

Apache Geronimo

Apache Geronimo is an open source server runtime that integrates the best open source projects to create Java/OSGi server runtimes that meet the needs of enterprise developers and system administrators.

Apache HttpClient

Although the `java.net` package provides basic functionality for accessing resources via HTTP, it doesn't provide the full flexibility or functionality needed by many applications. HttpClient seeks to fill this void by providing an efficient, up-to-date, and feature-rich package implementing the client side of the most recent HTTP standards and recommendations.

Designed for extension while providing robust support for the base HTTP protocol, HttpClient may be of interest to anyone building HTTP-aware client applications such as web browsers, web service clients, or systems that leverage or extend the HTTP protocol for distributed communication.

Apache HttpComponents

The Apache HttpComponents™ project is responsible for creating and maintaining a toolset of low level Java components focused on HTTP and associated protocols.

Within the HttpComponents project, [HttpCore](#) is a set of low level HTTP transport components that can be used to build custom client and server side HTTP services with a minimal footprint. HttpCore supports two I/O models: blocking I/O model based on the classic Java I/O and non-blocking, event driven I/O model based on Java NIO

Apache Log4j

Apache Log4j 2 is an upgrade to Log4j that provides significant improvements over its predecessor, Log4j 1.x, and provides many of the improvements available in Logback while fixing some inherent problems in Logback's architecture.

Apache Lucene, SOLR

The Apache Lucene™ project develops open-source search software.

Apache Tomcat, Tomcat Embed

Apache Tomcat is an open source software implementation of the Java Servlet and JavaServer Pages technologies.

Apache XBean :: Spring

XBean :: Spring provides a schema-driven proprietary namespace handler for Spring contexts.

Apache Xerces

The Apache Xerces Project is responsible for software licensed to the Apache Software Foundation intended for the creation and maintenance of:

- XML parsers
- related software components

Apache XML

The Apache XML Project used to be the home for many XML-related subprojects, many of which have moved to top-level project status recently or are currently in migration. The Apache XML Project slowly transforms into an place where you can find pointers to XML-related projects here in The Apache Foundation.

AspectJ

AspectJ is a seamless aspect-oriented extension to the Java programming language. It is Java platform compatible easy to learn and use.

AWS SDK for Amazon SQS

The AWS Java SDK for Amazon SQS module holds the client classes that are used for communicating with Amazon Simple Queue Service.

AWS SDK for Java Core

The AWS SDK for Java - Core module holds the classes that are used by the individual service clients to interact with Amazon Web Services. Users need to depend on aws-java-sdk artifact for accessing individual client classes.

Byte Buddy

Byte Buddy is a code generation and manipulation library for creating and modifying Java classes during the runtime of a Java application and without the help of a compiler.

CDI APIs

APIs for CDI (Contexts and Dependency Injection for Java).

cglib

cglib is a powerful, high performance and quality Code Generation Library, It is used to extend JAVA classes and implements interfaces at runtime.

DITA-OT

The DITA Open Toolkit is a Java-based implementation of the OASIS DITA Technical Committee's specification for DITA DTDs and schemas. It contains ANT, SAXON,...

DockPanel Suite

.Net Docking Library for Windows Forms

dom4j

dom4j is an easy to use, open source library for working with XML, XPath and XSLT on the Java platform using the Java Collections Framework and with full support for DOM, SAX and JAXP.

dsinfo

The dsinfo library enables you to easily use Scala-side information in implementations of embedded (internal) domain-specific languages. dsinfo is implemented using Scala macros which are an experimental feature of Scala 2.10 and 2.11.

dsprofile

The dsprofile library provides general facilities to implement domain-specific profiling in Scala and Java programs.

edtFTPj/Free

Free Java FTP library gives Java developers extensive FTP functionality.

Elasticsearch RESTful client

A RESTful client for the Elasticsearch search engine.

Fast Serialization

Fast Serialization reimplements Java Serialization with focus on speed (up to 10 times faster), size and compatibility. This allows the use of FST with minimal code change.

Fonto Editor

Fonto is an online XML editor designed for people with no knowledge of XML or any other technology that comes with structured content authoring.

GeckoFX

Gecko is a free and open source layout engine used in many applications developed by the Mozilla Foundation and the Mozilla Corporation (notably the Firefox web browser).

globalize

JavaScript globalization and localization. Formats and parses strings, dates and numbers in over 350 cultures.

GNU Aspell

GNU Aspell is a Free and Open Source spell checker designed to eventually replace Ispell. It can either be used as a library or as an independent spell checker. Its main feature is that it does a superior job of suggesting possible replacements for a misspelled word than just about any other spell checker out there for the English language. Unlike Ispell, Aspell can also easily check documents in UTF-8 without having to use a special dictionary. Aspell will also do its best to respect the current locale setting. Other advantages over Ispell include support for using multiple dictionaries at once and intelligently handling personal dictionaries when more than one Aspell process is open at once.

Specifically we are using GNUASpell dictionaries for de-CH, de-DE, en-CA, en-GB, en-US, es-ES, fr-FR, fr-CH, nl-NL.

google-code-prettify

google-code-prettify is a Javascript module and CSS file that allows syntax highlighting in an html page.

google-gson

google-gson is a Java library to convert JSON to Java objects and vice-versa.

Google Guava

The Guava project contains several of Google's core libraries that we rely on in our Java-based projects: collections, caching, primitives support, concurrency libraries, common annotations, string processing, I/O, and so forth.

GraphQL-Java

The Java implementation of GraphQL.

Hibernate

Hibernate is a high-performance Object/Relational persistence and query service. The most flexible and powerful Object/Relational solution on the market, Hibernate takes care of the mapping from Java classes to database tables and from Java data types to SQL data types. It provides data query and retrieval facilities that significantly reduce development time. Hibernate's design goal is to relieve the developer from 95% of common data persistence-related programming tasks by eliminating the need for manual, hand-crafted data processing using SQL and JDBC.

HK2 Framework

HK2 is a light-weight and dynamic dependency injection framework.

HSQldb (HyperSQL DataBase)

HSQldb (HyperSQL DataBase) is the leading SQL relational database engine written in Java. It offers a small, fast multithreaded and transactional database engine with in-memory and disk-based tables and supports embedded and server modes. It includes a powerful command line SQL tool and simple GUI query tools.

Hunspell

Hunspell is the spell checker of LibreOffice, OpenOffice.org, Mozilla Firefox 3 & Thunderbird, Google Chrome, and it is also used by proprietary software packages, like Mac OS X, InDesign, MemoQ, Opera and SDL Trados Studio.

InstallAnywhere

InstallAnywhere is the leading multi-platform development solution for application producers who need to deliver a professional and consistent cross installation experience for physical, virtual and cloud environments. From a single project file and build environment, InstallAnywhere creates reliable installations for on-premises platforms - Windows, Linux, Apple OS X, Solaris, AIX , HP-UX, and IBM iSeries - and enables you to take existing and new software products to a virtual and cloud infrastructure.

Jackson tooling

Inspired by the quality and variety of XML tooling available for the Java platform (StAX, JAXB, etc.), the Jackson is a multi-purpose Java library for processing JSON data format. Jackson aims to be the best possible combination of fast, correct, lightweight, and ergonomic components for developers.

JavaBeans Activation Framework

With the JavaBeans Activation Framework standard extension, developers who use Java technology can take advantage of standard services to determine the type of an arbitrary piece of data, encapsulate access to it, discover the operations available on it, and to instantiate the appropriate bean to perform said operation(s).

JavaBeans Validation

Bean Validation (JSR-303) API.

Javassist (*Java Programming Assistant*)

Javassist (*Java Programming Assistant*) makes Java bytecode manipulation simple. It is a class library for editing bytecodes in Java; it enables Java programs to define a new class at runtime and to modify a class file when the JVM loads it. Unlike other similar bytecode editors, Javassist provides two levels of API: source level and bytecode level. If the users use the source-level API, they can edit a class file without knowledge of the specifications of the Java bytecode. The whole API is designed with only the vocabulary of the Java language. You can even specify inserted bytecode in the form of source text; Javassist compiles it on the fly. On the other hand, the bytecode-level API allows the users to directly edit a class file as other editors.

javax.annotation

JSR 250 Common Annotations For The Java Platform.

javax.cache

Caching Java API

Javax Expression Language

Expression Language Java API

javax.inject

Dependency Injection Java API

JAXB

The goal of the JAXB project is to develop and evolve the code base for the Reference Implementation (RI) of JAXB, the Java Architecture for XML Binding. The JAXB specification is developed through the Java Community Process following the process described at jcp.org. This process involves an Expert Group with a lead that is responsible for delivering the specification, a reference implementation (RI) and a Technology Compatibility Kit (TCK). The primary goal of an RI is to support the development of the specification and to validate it. Specific RIs can have additional goals; the JAXB RI is a production-quality implementation that is used directly in a number of products by Oracle and other vendors.

JBoss Java Annotation Indexer (Jandex)

A Java Annotation Indexer for JBoss

JBoss Logging Framework

The JBoss Logging Framework.

jedis

A blazingly small and sane Redis Java client.

Jersey RESTful WS

Developing RESTful Web services that seamlessly support exposing your data in a variety of representation media types and abstract away the low-level details of the client-server communication is not an easy task without a good toolkit. In order to simplify development of RESTful Web services and their clients in Java, a standard and portable JAX-RS API has been designed. Jersey RESTful Web Services framework is open source, production quality, framework for developing RESTful Web Services in Java that provides support for JAX-RS APIs and serves as a JAX-RS (JSR 311 & JSR 339) Reference Implementation.

Jettison

Jettison is a collection of Java APIs (like STaX and DOM) which read and write JSON. This allows nearly transparent enablement of JSON based web services in services frameworks like CXF or XML serialization frameworks like XStream.

Jetty

The Jetty Web Server provides an HTTP server and Servlet container capable of serving static and dynamic content either from a standalone or embedded instantiations. Starting from Jetty version 7, the Jetty webserver and other core components are hosted by the Eclipse Foundation.

JLine

JLine is a Java library for handling console input. It is similar in functionality to BSD editline and GNU readline. People familiar with the readline/editline capabilities for modern shells (such as bash and tcsh) will find most of the command editing features of JLine to be familiar.

JMESPath Java

JMESPath is a query language for JSON. You can extract and transform elements from a JSON document. This is a Java implementation

Joda-Convert

Joda-Convert provides a small set of classes to provide round-trip conversion between Objects and Strings. It does not tackle the wider problem of Object to Object transformation.

Joda-Time

Joda-Time provides a quality replacement for the Java *date* and *time* classes. The design allows for multiple *calendar* systems, while still providing a simple API. The 'default' calendar is the http://www.joda.org/joda-time/cal_iso.html standard which is used by XML. The Gregorian, Julian, Buddhist, Coptic, Ethiopic and Islamic systems are also included, and we welcome further additions. Supporting classes include time zone, duration, format and parsing.

jQuery

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

jquery-cookie

jQuery plugin for reading, writing and deleting cookies.

jquery.datatables

DataTables is a plug-in for the jQuery Javascript library. It is a highly flexible tool, based upon the foundations of progressive enhancement, which will add advanced interaction controls to any HTML table.

jquery.dataTables.columnFilter

Adds advanced filter capabilities to the DataTables. JS file.

jQueryFileUpload

File Upload widget with multiple file selection, drag&drop support, progress bar, validation and preview images, audio and video for jQuery.

jquery.TypeScript.DefinitelyTyped

TypeScript Definitions (d.ts) for jquery.

jQuery Highlight

Highlights the search keywords/terms in a preview.

jQuery UI

jQuery UI is a set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library.

JSON-js

JSON is a light-weight, language independent, data interchange format. See <http://www.JSON.org/> / The files in this collection implement JSON encoders/decoders in JavaScript. JSON became a built-in feature of JavaScript when the ECMAScript Programming Language Standard - Fifth Edition was adopted by the ECMA General Assembly in December 2009. Most of the files in this collection are for applications that are expected to run in obsolete web browsers. For most purposes, json2.js is the best choice.

Json.NET

Json.NET is a popular high-performance JSON framework for .NET.

JTA (Java Transaction API)

The `javax.transaction` package. It is appropriate for inclusion in a classpath, and may be added to a Java 2 installation.

Kiama

The Kiama test library contains a collection of examples, tests that use those examples, and useful test support code.

Knockout JavaScript library

Knockout is a JavaScript library that helps you to create rich, responsive display and editor user interfaces with a clean underlying data model. Any time you have sections of UI that update dynamically (e.g., changing depending on the user's actions or when an external data source changes), KO can help you implement it more simply and maintainably.

kXML 2

kXML is a small XML pull parser, specially designed for constrained environments such as Applets, Personal Java or MIDP devices. In contrast to kXML 1, kXML 2 is based on the XML pull API.

Logback

Logback is intended as a successor to the popular log4j project, picking up where log4j leaves off.

MVC Web Projects

Auxiliary MVC Web Project libraries.

MXP1

MXP1 is a stable XmlPull parsing engine that is based on ideas from XPP and in particular XPP2 but completely revised and rewritten to take the best advantage of latest JIT JVMs such as Hotspot in JDK 1.4+.

Objenesis

Objenesis is a small Java library that serves one purpose: to instantiate a new object of a particular class.

NHunspell

NHunspell brings the spell checking, hyphenation and thesaurus to the Microsoft® .NET Framework. NHunspell is C# library and wraps native libraries for Hunspell, Hyphen and MyThes. One design goal of this library and wrapper is to keep the source code of the included libraries as unmodified as possible. New versions of the base libraries can therefore easily adopted to NHunspell.

The integrated libraries are used in OpenOffice and they work with the dictionaries published on OpenOffice.org.

NLog

NLog is a free logging platform for .NET, Silverlight and Windows Phone with rich log routing and management capabilities. NLog makes it easy to produce and manage high-quality logs for your application regardless of its size or complexity.

okhttp

An HTTP+HTTP/2 client for Android and Java applications.

okio

A modern I/O API for Java.

PATRICIA Trie in Java

An implementation of the Practical Algorithm to Retrieve Information Coded in Alphanumeric (PATRICIA).

Postal.Mvc5

Generate emails using ASP.NET MVC views

PS Cmdlet Help Editor

PowerShell Cmdlet Help Editor is the tool that helps you to create and edit XML-based help files for your PowerShell modules and PSSnap-Ins.

Red Hat Linux

Red Hat Enterprise Linux OpenStack Platform delivers an integrated foundation to create, deploy, and scale a secure and reliable public or private OpenStack cloud. Red Hat Enterprise Linux OpenStack Platform combines the world's leading enterprise Linux and the fastest-growing cloud infrastructure platform to give you the agility to scale and quickly meet customer demands without compromising on availability, security, or performance.

Rx .NET

Reactive Extensions for .NET library used to validate entered values

Scallop

Scallop is a command line parser.

Scala

The Scala programming language fuses object-oriented and functional programming in a statically typed programming language. It is aimed at the construction of components and component systems.

SitemapGen4j

SitemapGen4j is a library to generate XML sitemaps in Java.

SLF4J

The Simple Logging Facade for Java (SLF4J) serves as a simple facade or abstraction for various logging frameworks (e.g. java.util.logging, logback, log4j) allowing the end user to plug in the desired logging framework at deployment time.

SnakeYAML

YAML is a data serialization format designed for human readability and interaction with scripting languages. SnakeYAML is a YAML parser and emitter for the Java programming language.

SNMP4J

SNMP4J is an enterprise class free open source and state-of-the-art SNMP implementation for Java™ 2SE 1.4 or later. SNMP4J supports command generation (managers) as well as command responding (agents). Its clean object oriented design is inspired by SNMP++, which is a well-known SNMPv1/v2c/v3 API for C++.

SpringFox

Automated JSON API documentation for API's built with Spring.

Spring Framework

The Spring Framework provides a comprehensive programming and configuration model for modern Java-based enterprise applications - on any kind of deployment platform. A key element of Spring is infrastructural support at the application level: Spring focuses on the "plumbing" of enterprise applications so that teams can focus on application-level business logic, without unnecessary ties to specific deployment environments.

StAX

StAX is a standard XML processing API that allows you to stream XML data from and to your application. This StAX implementation is the standard pull parser implementation for JSR-173 specification.

Swagger

Swagger is a simple yet powerful representation of your RESTful API. With the largest ecosystem of API tooling on the planet, thousands of developers are supporting Swagger in almost every modern programming language and deployment environment. With a Swagger-enabled API, you get interactive documentation, client SDK generation and discoverability.

Swashbuckle.Core

Seamlessly adds a Swagger to WebApi projects.

Thinkecture IdentityServer

Front-end Secure Token Service to serve SAML tokens.

TwelveMonkeys Common

TwelveMonkeys Common library contains common utility classes relating to languages, I/O and images.

TwelveMonkeys ImageIO

TwelveMonkeys ImageIO is a collection of plugins and extensions for Java's ImageIO. These plugins extends the number of image file formats supported in Java, using the `javax.imageio.*` package. The main purpose of this project is to provide support for formats not covered by the JRE itself.

ua-parser

A multi-language port of Browserscope's user agent parser.

Xalan-Java

Xalan-Java is an XSLT processor for transforming XML documents into HTML, text, or other XML document types. It implements XSL Transformations (XSLT) Version 1.0 and XML Path Language (XPath) Version 1.0 and can be used from the command line, in an applet or a servlet, or as a module in other program.

Thinkecture IdentityServer

Front-end Secure Token Service to serve SAML tokens.

WiX

The WiX toolset builds Windows installation packages from XML source code. The tool-set integrates seamlessly into build processes.

Woodstox

Woodstox is a high-performance validating namespace-aware StAX-compliant (JSR-173) Open Source XML-processor written in Java.

XML Pull Parsing

An XML Pull Parsing API.

XStream

XStream is a simple library to serialize objects to XML and back again.

XULRunner

XULRunner is a runtime environment developed by the Mozilla Foundation to provide a common back-end for previewing.