Content Composer

Installation Guide

Foundation 23.1

Written by: Documentation Team, R&D
Date: Thursday, February 9, 2023

Hyland

Documentation Notice

Information in this document is subject to change without notice. The software described in this document is furnished only under a separate license agreement and may only be used or copied according to the terms of such agreement. It is against the law to copy the software except as specifically allowed in the license agreement. This document or accompanying materials may contain certain information which is confidential information of Hyland Software, Inc. and its affiliates, and which may be subject to the confidentiality provisions agreed to by you.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright law, no part of this document 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 Hyland Software, Inc. or one of its affiliates.

Hyland, HXP, OnBase, Alfresco, Nuxeo, and product names are registered and/or unregistered trademarks of Hyland Software, Inc. and its affiliates in the United States and other countries. All other trademarks, service marks, trade names and products of other companies are the property of their respective owners.

© 2023 Hyland Software, Inc. and its affiliates.

The information in this document may contain technology as defined by the Export Administration Regulations (EAR) and could be subject to the Export Control Laws of the U.S. Government including for the EAR and trade and economic sanctions maintained by the Office of Foreign Assets Control as well as the export controls laws of your entity’s local jurisdiction. Transfer of such technology by any means to a foreign person, whether in the United States or abroad, could require export licensing or other approval from the U.S. Government and the export authority of your entity’s jurisdiction. You are responsible for ensuring that you have any required approvals prior to export.

Table of Contents

Overview 7
Technical Specifications *7*
Updating Content Composer *7*
Installing Content Composer 7
Content Composer Installation *7*
Installation Package 7
Content Composer License 8
MS SQL Server Prerequisites *8*
Oracle Prerequisites *8*
Installing ODP.NET Driver for Content Composer Server and Studio 8
Verifying the OracleODP.NET Installation 9
VerifyOra: Troubleshooting Error Messages 10
About the Environment Variable ComposerDir *11*
About RSA Encryption *11*
Installing the First Content Composer Server *11*
Installing an Additional Content Composer Server *12*
Performing a Complete Installation of Content Composer *12*
Creating the RSA Container *12*
Verifying an RSA Container *13*
Granting Access Rights to the RSA Container *13*
Creating the Oracle Users *14*
Creating the Database Roles in SQL Server *14*
Creating the Database Users in SQL Server *14*
Configuring Content Composer Server *15*
Exporting and Backing up Encryption Keys *18*
Importing the RSA Encryption Keys *19*
Permission Configuration (Content Composer Server) *19*
Confirming Composer Word Addin Installation (Content Composer Studio) *19*
About HTTP or HTTPS for the MWS SOAP API Communication Channel *19*
Enabling HTTPS for the MWS SOAP API Communication Channel 20
Enabling HTTP for MWS SOAP API Communication Channel 20

About Configuring the MWS REST API Communication Channel *23*
Enabling HTTPS for the MWS REST API Communication Channel 23
Enabling HTTP for the MWS REST API Communication Channel 24
Checking the Availability of the MWS REST API 25
About Configuring Hyland IdP *25*
Prerequisites 25
Configuring Content Composer IdP Authentication 26
Modifying the IdP Configuration File 27
Testing the IdP Configuration 27
Configuring a Content Composer Server Installation to Use Hyland IdP 28
Configuring a Content Composer Studio Installation to Use Hyland IdP 30
Content Composer Windows Services and Console Applications with Hyland IdP 32
About the Secrets File 32
Creating the Secrets File 33
Configuring Content Composer Windows Services and Console Applications to Use Hyland IdP 33
Configuring Content Composer Windows Service Authorization for IdP Authentication Type Client Credentials 36
About Configuring Content Composer Web Client to Use Hyland IdP 37
Installing Content Composer Studio only 37
Installing Content Composer Studio *37*
Importing the RSA Encryption Keys *38*
Granting Access Rights to the RSA Container *38*
Configuring Content Composer Studio *38*
Confirming Composer Word Add-In Installation *39*
Starting the Core Service *39*
Launching Content Composer Studio *39*
Installing Content Composer Client only 39
Installing Content Composer Client *40*
Configuring Content Composer Client *40*
User Language 41
Installation Packages for Additional Languages (Content Composer Studio and Client) *41*
Installing a Language Package (Content Composer Studio and Client) *41*

Setting the Language for the Odin Views (Content Composer Server) *41*
Running Content Composer in Another Language *42*
Unattended Installation and Uninstallation of a Language Package *42*
Installing a Language Package Unattended 43
Uninstalling a Language Package Unattended 43
Unattended Installation and Uninstallation of Content Composer 43
Installing Content Composer Unattended *45*
Uninstalling Content Composer Unattended *45*
Advanced Installation 45
Odin or MWS Database Setup *45*
MS SQL Server *46*
Oracle *46*
Setting up the Database *46*
Installation of Odin or MWS Manually on an Oracle Database (Composer Server) *48*
Database Schema 49
Installation of Odin or MWS Manually on a MS SQL Server Database (Composer Server) *49*
Database Schema 49
Notes 49
Database User 49
Repository Setup (Content Composer Server) *49*
License 50
Oracle 50
Installing the Repository (Content Composer Server) *50*
Modification of the Server Configuration *52*
Modify UserRepository.config (Content Composer Server) 52
Modify Composer.MWS.exe.config (Content Composer Server) 52
Specify Normal.dotm (Content Composer Server) 52
Modify Composer.Client.exe.config (Content Composer Client) 52
Example 53
Start the Core service (Content Composer Studio) 53
Launching Content Composer Studio *53*
Entering Connection Data for the Odin or MWS Database in the Configuration (Content Composer Server) *54*

Starting the Services *54*
Troubleshooting 55
Content Composer Word Add-In *55*
RSA *55*
Deleting an RSA Container *55*

Overview

This document contains information on how to install Content Composer and Content Composer language packs on your system.

The following types of installation are available.

- **Content Composer Server**. The server component of Content Composer.
- **Content Composer Studio**. The design environment of Content Composer.
- **Content Composer Client**. The client component of Content Composer.

Technical Specifications

For information on the technical specifications for this application, see the *Content Composer Technical Specifications Guide*.

Updating Content Composer

For information on how to update an existing Content Composer installation , see the *Content Composer Update Guide*.

Installing Content Composer

You can perform the setup of Content Composer using one of the following options.

- To execute an unattended installation, see the Unattended Installation and Uninstallation of Content Composer section.
- To install only the Content Composer Studio component, see the Installing Content Composer Studio only section.
- To install only the Content Composer Client component, see the Installing Content Composer Client only section.
- To perform the complete installation of Content Composer on your system, see the Content Composer Installation section.

Content Composer Installation

This section provides information on the installation of Content Composer.

Installation Package

- **Content Composer Setup 23.1.0 for Windows (64bit).exe**.

  This installs Content Composer on a 64-bit Windows OS as a native 64-bit application. The drivers used by Content Composer to access other systems must be installed in the 64-bit version.

Content Composer License

- If you do not have a valid Content Composer license, email Hyland Support and request a license for your environment.
- Copy the license file to the Content Composer installation directory and ensure that the file name is **Composer.lic**.

MS SQL Server Prerequisites

If you are using Microsoft SQL Server as the database for Content Composer, consider the following prerequisites.

- An empty database for the Odin/MWS DB schemas.
- An empty database for the repository DB schema.
- The SQL Server authentication mode must be active.

Oracle Prerequisites

Content Composer requires the Oracle ODP.NET version 19c to access an Oracle DB.

Before you install Content Composer, follow the instructions in the following topics.

- Installing ODP.NET Driver for Content Composer Server and Studio
- Verifying the OracleODP.NET Installation

Installing ODP.NET Driver for Content Composer Server and Studio

To install the ODP.NET driver, complete the following steps:

1. To perform the installation using the XCOPY installer, complete the following substeps:
    1. Go to https://www.oracle.com/database/technologies/dotnet-odacdev-downloads.htm and locate the **ODAC XCopy** section.
    2. Download **64-bit ODAC 19.3**.
    3. Unzip the downloaded file **ODAC193Xcopy_x64.zip**.
    4. Open a command prompt window as an administrator and navigate to the directory that contains the contents of the **64-bit ODAC 19.3** ZIP file.
    5. Execute the following command.

        ```
        install.bat odp.net4 [oracle installation directory][oracle home key] true true
        ```

        Example

        ```
        install.bat odp.net4 c:\oracle odac true true
        ```

2. To perform the installation using the Oracle Universal Installer (OUI), complete the following substeps:
    1. Go to to https://www.oracle.com/database/technologies/dotnet-odacdev-downloads.html and locate the **ODAC OUI** section.
    2. Download **64-bit ODAC 19.3.1**.

3. Unzip the downloaded file and then run the **setup.exe** file included in the ZIP file.
4. On each page, provide the required information and then click **Next**.

   Note: In the **Available Product Components** page, select only **Oracle Data Provider for .NET**. Other components are not required.

5. Review the summary and then click **Install**.

3. Add the **ODP.NET** installation directory and its **bin** subdirectory to the Windows environment variable `PATH`, if not yet present.

   Example

   If `C:\Oracle` is the installation directory, add `C:\Oracle` and `C:\Oracle\bin`.

   Note:Ensure to place the newly added directories before all other Oracle directories.

4. To register the **Oracle.DataAccess.dll** in the Global Assembly Cache (GAC), complete the following substeps:
   1. Open a command prompt window as an administrator and navigate to the directory that contains the **OraProvCfg.exe** file.

      Note: The default directory is `[Oracle installation directory]\odp.net\bin\4`.

   2. Execute the following command.

      ```
      oraprovcfg /action:gac /providerpath:[full path to Oracle.DataAccess.dll]
      ```

      Example

      ```
      oraprovcfg /action:gac
      /providerpath:C:\oracle\odp.net\bin\4\Oracle.DataAccess.dll
      ```

Verifying the OracleODP.NET Installation

To verify that the Oracle ODP.NET is installed successfully on your system, complete the following steps:

1. From the Content Composer setup directory, copy the following files to a temporary directory.
   - VerifyOra_64.bit.exe
   - VerifyOra_64.bit.exe.config
2. Open a **Command Prompt** window and navigate to the directory where you copied the files in the previous step.
3. Type `VerifyOra_64bit.exe` and then press **Enter**.
4. Review the output.

   The VerifiyOra tool writes either **SUCCESS** or an error message to the Command Prompt window and to the log file **VerifyOra.txt**, located in the **My Documents** directory of the executing Windows user account. In case of an error message, see VerifyOra: Troubleshooting Error Messages.

5. To verify the connection to an Oracle database server, execute the following command.

Note: You can execute the command using any existing Oracle user.

```
VerifyOra_64bit.exe "[Oracle server name]" "[Oracle user name]" "
[password]"
```

Example

```
VerifyOra_64bit.exe "MyOra:1521/orcl" "alincoln" "password"
```

VerifyOra: Troubleshooting Error Messages

The following table displays the error messages and their solutions.

Error messages	Solution
The program cannot start because MSVCR120.dll is missing from your computer. Try reinstalling the program to fix this problem.	• Install **Visual C++ Redistributable Packages for Visual Studio 2013**. Download the 64-bit version from www.microsoft.com
The procedure entry point ons_error_set could not be located in the dynamic link library oraons.dll	• If you have multiple Oracle clients installed, check your `PATH` environment variable • Reorder the entries for Oracle so that when you navigate from beginning to end of your path, you start from the current Oracle clients.
Could not load file or assembly 'Oracle.DataAccess, Version=4.122.19.1, Culture=neutral, PublicKeyToken=89b483f429c47342' or one of its dependencies. The system cannot find the file specified	• Check whether the Oracle client is installed. • Check the version of the installed Oracle client.
Unable to load DLL 'OraOps19.dll': The specified module could not be found. (Exception from HRESULT: 0x8007007E.	• Check that the environment variable `PATH` contains the path of the Oracle client installation directory. For more information, see Installing ODP.NET Driver for Content Composer Server and Studio. • Install **Visual C++ Redistributable Packages for Visual Studio 2013**. Download the 64-bit version from www.microsoft.com.

About the Environment Variable *ComposerDir*

The Content Composer setup creates the environment variable **ComposerDir**, which contains the path to the Content Composer installation directory.

ComposerDir is a system variable and as such visible to all Microsoft Windows users.

Example

If Content Composer was installed in the directory *D:\ContentComposer*, the environment variable has the value *D:\ContentComposer*.

Note: The value of the environment variable ends with a backslash.

About RSA Encryption

To improve security, Content Composer generates random encryption keys during installation.

The Content Composer setup includes the tool **Composer.EncryptionTool**, which creates the new keys and updates existing encrypted data.

Content Composer uses these keys to encrypt all passwords stored in the Content Composer database and configuration files.

The following topics provide instructions for backing up and sharing these keys between Content Composer servers.

Caution

All Content Composer server and Studio installations must use the ***same RSA key***. Use the ***same RSA key*** for all your different environments, such as test and production environments.

Installing the First Content Composer Server

To install the first Content Composer server in your IT environment, complete the following tasks:

1. Install Content Composer server. See Performing a Complete Installation of Content Composer for more information.
2. Create the RSA container. See Creating the RSA Container for more information.
3. Verify that the RSA container has been successfully created. See Verifying an RSA Container for more information.
4. Grant access rights to the RSA container. See Granting Access Rights to the RSA Container for more information.
5. Create the database users and roles. See Creating the Oracle Users or Creating the Database Roles in SQL Server and Creating the Database Users in SQL Server for more information.
6. Configure the Content Composer server. See Configuring Content Composer Server for more information.
7. Export the encryption keys. See Exporting and Backing up Encryption Keys for more information.

Installing an Additional Content Composer Server

To install an additional Content Composer server, complete the following tasks:

1. Install Content Composer server. See Performing a Complete Installation of Content Composer for more information.
2. Import the RSA encryption keys. See Importing the RSA Encryption Keys for more information.
3. Grant access rights to the RSA container. See Granting Access Rights to the RSA Container for more information.

Performing a Complete Installation of Content Composer

To perform a complete installation of Content Composer, complete the following steps:

1. In **Windows File Explorer**, right-click **Content Composer Setup 23.1.0 for Windows (64bit).exe**, and then click **Properties**.
2. If the **Unblock** check box is displayed at the bottom right of the **Content Composer Setup 23.1.0 for Windows (64bit).exe**, dialog box, select **Unblock**.

   Note: If the **Unblock** check box is not displayed, the file is already unblocked.
3. Click **OK**.
4. Double-click the **Content Composer Setup 23.1.0 for Windows (64bit).exe** file.
5. Optional. In the dialog box, click **Install** if Microsoft Visual Studio Tools for Office Runtime tool is not installed on your system.
6. In the **Welcome to the Installation Wizard for Content Composer** page, click **Next**.
7. In the**License Agreement** page, review the terms in the License Agreement, scroll to the end of the agreement, click **I accept the terms in the license agreement**, and click **Next**.
8. In the **Destination Folder** page, accept the default directory, and click **Next** or click **Change** to select an alternate directory.
9. In the **Setup Type** page, click **Complete** to install all Content Composer features or click **Custom** to choose the Content Composer features you want to install.
10. If you want to choose the Content Composer features to install, in the **Custom Setup** page, select one or more of the following Content Composer features.
    - Content Composer Server. Installs the server component of Content Composer.
    - Content Composer Studio. Installs the complete design environment of Content Composer.
    - Content Composer Client. Installs the client component of Content Composer.
11. In the **Ready to Install the Program** page, click **Install**.
12. In the **Content Composer Wizard Completed** page, click **Finish**.

Creating the RSA Container

After installing your first Content Composer server, you need to create an exportable RSA container. Complete the following steps:

1. Open a command prompt window as an administrator and navigate to the **%Composerdir%**

directory.

2. Execute the following command:

```
Composer.EncryptionTool create-rsa
```

Verifying an RSA Container

To verify if an RSA container exists on the computer:

1. Open a command prompt window as an administrator and navigate to the **%ComposerDir%** directory.
2. Execute the following command:

```
Composer.EncryptionTool info
```

- If the RSA key exists, the following output is displayed:

```
dd.mm.yyyy hh:mm;Checking if the RSA key container already
exist in the machine.
dd.mm.yyyy hh:mm;searching for RSA key container with name
'CoCo' ....
dd.mm.yyyy hh:mm;Printing information about rsa container
dd.mm.yyyy hh:mm;RSA key container file path: [File-Path]
dd.mm.yyyy hh:mm;the RSA key container is exportable
found RSA key container with name 'CoCo'!
End of execution
```

- If the message `the RSA key container is not exportable` is displayed, complete the steps in the following topics:
    1. Deleting an RSA Container
    2. Creating the RSA Container

Granting Access Rights to the RSA Container

To grant access rights to the RSA container to every Windows account that runs a Content Composer service or Studio, complete the following steps:

1. Open a command prompt window as an administrator and navigate to the **%windir%\Microsoft.NET\Framework64\v4.0.30319** directory.
2. Execute the following command, which grants access rights to the user `CindySmith`, who is a member of the domain `OnBase`, replacing domain and user name with the required values:

```
aspnet_regiis -pa coco "onbase\cindysmith"
```

3. Repeat the previous step for each Windows account running a Content Composer service or Studio.

Creating the Oracle Users

Before you set up the Oracle database, you need to create the required users.

Prerequisite

You must have **SYSDBA** permissions to run the scripts.

To create the Oracle users, complete the following steps:

Note: The scripts are located in the **%Composerdir%admin\Oracle** directory.

1. Execute the **Ora_Odin_CreateUser.sql** script against the database in which the Odin/MWS database schema is to be installed.
2. Execute the **Ora_Rep_CreateUser.sql** script against the database in which the repository database schema is to be installed.

Creating the Database Roles in SQL Server

To improve security, Content Composer provides both read-only and full-access roles for the repository, MWS and Odin databases.

To create the database roles, complete the following steps:

1. From the **%Composerdir%admin\MS-SQL** directory, open the **MSSql_CreateRole.sql** script.
2. Follow the instructions provided at the top of the script.

   In the **Specify Values for Template Parameters** dialog box, specify the parameters for the **Repository** roles.

   The script creates the following roles.

   - **composer_full**: Provides full access.
   - **composer_readonly**: Provides read-only access.
3. Repeat the previous steps for both the **MWS** and the **Odin** roles.

Creating the Database Users in SQL Server

To create the database users, complete the following steps:

1. From the **%Composerdir%admin\MS-SQL** directory, open the **MSSql_CreateUser.sql** script.
2. Follow the instructions provided at the top of the script.

   In the **Specify Values for Template Parameters** dialog box, specify the parameters for the **Repository** user that you want to assign to the **composer_full** full role.
3. Create another user for the **Repository** database and assign this user to the **composer_readonly** role.
4. Repeat the previous steps for both the **MWS** and the **Odin** users.

Configuring Content Composer Server

To configure your Content Composer server and install the required databases, complete the following steps:

1. In the installation directory, double-click **Composer.SetupAssistant.exe** file and on the **Content Composer Setup** tab, click **Next**.
2. On the **Content Composer License** page, select the Content Composer license file.
3. On the **Content Composer Database** page, click **Next**. The Content Composer Database Setup page launches to install the repository database.
4. On the **Content Composer Database Setup** page, verify the Content Composer Repository installation prerequisites. Each step in the list must show OK before you can proceed to the next page. In case of errors, resolve all indicated errors and click **Evaluate**.
5. On the **Select Procedure** page, click **Next** and in the **Database Connection** page, select one of the following options and then click **Edit Connection**.

Option	Description
MS SQL Server - Repository database	On the **SQL Server Connection** page, complete the following steps: 1. In the **Enter or select server name** box, type the name of the SQL Server instance. 2. In the **Server logon** box, complete only one of the following substeps: • Select the **Use SQL Server Authentication** check box. In the **User** box, type the name of the database user you want to use to create the repository database schema and in the **Password** box, type the password of the database user. **Note**: Use the database user created in Create the database users in SQL Server that has the **composer_full** role. • Select the **Use Windows authentication** check box. 3. In the **Database** box, type the name of the SQL Server database to which you want to install the repository database schema. 4. Click **Test Connection** and click **OK**.
Oracle – Repository database	On the **Oracle connection properties** page, complete the following steps: 1. In the **Enter server name** box, type the name of the Oracle server instance.

Option	Description
	2. In the **User** box, type the name of the database user you want to use to create the repository database schema. **Note**: Use the database user created by the `Ora_Rep_CreateUser.sql` script because the user must have specific permissions. 3. In the **Password** box, type the password of the database user. 4. Click **Test Connection** and click **OK**. 5. Click **Next** and in the **Creating the system** page, complete the following substeps: 1. In the **System Name** box, type the name of the Composer system you want to create and in the **SystemObjectID** box, type the ID of the system you want to create. 2. Optional. In the **Database Alias** box, change the predefined value.

6. Click **Next** and in the **Installation** page, click **Start installation**.
7. When the Content Composer Database setup is complete, click **OK**.
8. In the **Content Composer Odin / MWS Database** page, click **Next**. The **Content Composer Database Setup** page launches to install the Odin and MWS database.
9. On the **Composer Database Setup** page, verify the installation prerequisites. Each step in the list must show OK before you can proceed to the next page. In case of errors, resolve all indicated errors and click **Test**.
10. On the **Select Procedure** page, click **Next**.
11. On the **Database Connection** page, select one of the following database products and click **Edit Connection** and in the **Installation** page, click **Start installation**.

Option	Description
MS SQL Server - Odin/MWS database	On the **SQL Server Connection** page, complete the following steps: 1. In the **Enter or select server name** box, type the name of the SQL Server instance. 2. In the **Server logon** box, complete only one of the following substeps:

Option	Description
	• Select the **Use SQL Server Authentication** check box. In the **User** box, type the name of the database user you want to use to create the database schema. **Note**: Use the database user created in Create the database users in SQL Server that has the **composer_full** role. • Select the **Use Windows authentication** check box. 3. In the **Database** box, type the name of the SQL Server database to which you want to install the Odin and MWS database schema. 4. Click **Test Connection** and click **OK**.
Oracle - Odin/MWS database	On the **Oracle connection properties** page, complete the following steps: 1. In the **Enter server name** box, type the name of the Oracle server instance. 2. In the **User** box, type the name of the database user you want to use to create the Odin/MWS database schema. **Note**: Use the database user created by the `Ora_Odin_CreateUser.sql` script because the user must have specific permissions. 3. In the **Password** box, type the password of the database user. 4. Click **Test Connection** and click **OK**.

12. On the **Select Odin / MWS option** page, select **Create Odin and MWS schema** and click **Next**.
13. On the **Installation** page, click **Start installation**.
14. When the Odin/MWS database is complete, click **OK**.
15. On the **Select a Content Composer system** page, select the base system and click **Next**.
16. On the **Content Composer Software Configuration** page, to select the location of the Content Composer services, perform one of the following actions.
    - Select **Local**.

- Select **Specific workstation**, and in the **Computer name** box, type the computer name.

17. On the **Content Composer Microsoft Word Template** page, select the required Normal.dotm file and click **Next**.
18. On the **Configure the Content Composer database alias objects** page, to the right of the MWS connection string box, click **Edit**.
19. On the **Connection Settings** dialog box, complete the following substeps:
    1. Under **Data Provider**, select one of the following data providers.
       - Native SQL for MS SQL Server
       - Microsoft Oracle for an Oracle database
    2. Click the ellipsis button to the right of the Connection String box and in the **Connection Properties** dialog box, enter the connection information for the MWS database.
    3. Click **Test connection** and click **OK**.
20. On the **Configure the Content Composer database alias objects** page, to the right of the **Odin connection string** box, click **Edit**, repeat the previous steps, and click **Next**.
21. On the **Configure language of Odin** page, select the preferred language and click **Next**.
22. On the **Administrate the Content Composer services** page, select **Restart services** and click **Next**.
23. On the **Close the Content Composer Assistant** page, select one of the following options and click **OK**.
    - Restart the assistant
    - Close

Exporting and Backing up Encryption Keys

If this is your first installation of Content Composer in your IT environment, you need to export and backup your encryption keys. Complete the following steps:

1. Open a command prompt window as an administrator and navigate to the **%Composerdir%** directory.
2. Execute the following command,

   ```
   Composer.EncryptionTool exportkey
   ```

3. From the **%Composerdir%** directory, copy the following files to a **secure backup store**:

   Important: A backup copy of these files is of utmost importance, as the keys cannot be recovered if lost!
   - CoCoEncryptionRSAKey.xml
   - encryption_keys.config
4. From the **%Composerdir%** directory, delete the following file:
   - CoCoEncryptionRSAKey.xml

Importing the RSA Encryption Keys

If you are performing an additional installation of Content Composer components, you need to import the encryption keys created with your first Content Composer server installation. Complete the following steps:

1. From your secure backup store, copy the following files to the **%Composerdir%** directory:
   - encryption_keys.config
   - CoCoEncryptionRSAKey.xml
2. Open a command prompt window as an administrator and navigate to the **%Composerdir%** directory.
3. To import the encryption key, execute the following command:

   ```
   Composer.EncryptionTool importkey
   ```

4. For security reasons, delete `CoCoEncryptionRSAKey.xml` from the **%Composerdir%** directory after importing.

Permission Configuration (Content Composer Server)

In Content Composer Server, ensure that the following security setting for the Content Composer installation directory is set.

- The user group **Users** must be assigned **Full control**.

Confirming Composer Word Addin Installation (Content Composer Studio)

The Content Composer Setup delivers the Composer.WordAddIn. This add-in for Microsoft Word enables the user to insert text block variables directly into a Word document.

Most combinations of Microsoft Windows and Microsoft Word installations require confirmation of the Word AddIn installation. To confirm the installation of Composer.WordAddIn, complete the following steps:

1. Start Microsoft Word.
2. If the **Microsoft Office Customization Installer** dialog box appears, click **Install**.
3. On the toolbar, click **File > Options**.
4. In the left pane, click **Add-ins**. If **Composer.WordAddIn** is already installed on your system, **Content Composer Word AddIn** exists under **Active Application Add-ins**.

   Note: If there are issues in installing the Content Composer Word AddIn, complete the steps described in the Content Composer Word Add-In section.

About HTTP or HTTPS for the MWS SOAP API Communication Channel

By default, Content Composer uses HTTPS to encrypt the communication between third party applications and the MWS SOAP API endpoint.

You can also configure these communications to use HTTP.

Note: For security reasons, we recommend the use of HTTPS.

For information on how to configure HTTPS, see Enabling HTTPS for the MWS SOAP API Communication Channel.

For information on how to change the default HTTPS configuration to use the insecure HTTP protocol, see Enabling HTTP for MWS SOAP API Communication Channel.

Enabling HTTPS for the MWS SOAP API Communication Channel

To enable HTTPS, complete the following steps on the Content Composer server:

Prerequisite

An SSL certificate from a Certificate Authority.

This certificate must have a subject name or subject alternative name that matches the DNS name of your Content Composer server.

1. Import your SSL certificate into the certificate store under **Local Computer > Personal**.
2. Verify that **Certification Path** and **Certificate status** are correct.
3. To bind the SSL server certificate to IP port 8111, complete the following substeps:
    1. Open the certificate.
    2. On the **Details** tab, find the **Thumbprint** field and copy its value to the clipboard.
    3. Open a **Command Prompt** window with Administrator rights.
    4. Execute the following command, replacing `SSL-cert-thumbprint` with the thumbprint value of your certificate.

        Example

        ```
        netsh http add sslcert ipport=0.0.0.0:8111 certhash=SSL-cert-thumbprint appid={[any GUID]}
        ```
4. To reserve the URL endpoint for non-administrator users and accounts, complete the following substeps:
    1. Identify the Windows user account that is configured to execute the Windows service **Composer.MWS**.
    2. Open a **Command Prompt** window with Administrator rights.
    3. To reserve the MWS URL, execute the following command, replacing `Windows-user-account` with the user account used to execute the Windows service Composer.MWS.

        Example

        ```
        netsh http add urlacl url=https://+:8111/ user=Windows-user-account
        ```
    4. Restart all Content Composer services.

Enabling HTTP for MWS SOAP API Communication Channel

To enable HTTP, complete the following steps on the Content Composer server:

Note: For security reasons, we recommend the use of HTTPS.

1. From the Content Composer installation directory, open **Composer.MWS.exe.config** with a text

editor that supports UTF-8.

2. Find the following lines and comment them out.

```
<service name="ModusSuite.MWS.MWSProcessServiceBasic"
behaviorConfiguration ="BasicBehaviour_SSL">
...
</service>
```

Example

```
<!--
<service name="ModusSuite.MWS.MWSProcessServiceBasic"
behaviorConfiguration ="BasicBehaviour_SSL">
...
</service>
-->
```

3. Find the following line.

```
<service name="ModusSuite.MWS.MWSProcessServiceBasic"
behaviorConfiguration ="BasicBehaviour" >
```

4. Uncomment the **Insecure HTTP Protocol** part and comment out the **Secure HTTPS Protocol** part as shown in the following example.

Example

```
<!-- Insecure HTTP Protocol -->
<service name="ModusSuite.MWS.MWSProcessServiceBasic"
behaviorConfiguration="BasicBehaviour">
<host>
  <baseAddresses>
    <add baseAddress="http://localhost:8011/mwsbasic" />
  </baseAddresses>
</host>
<endpoint address="mwsprocess"
          binding="basicHttpBinding"
          bindingConfiguration="BasicBinding"
          contract="ModusSuite.MWS.Types.IMWSProcessServiceBasic" />
</service>
<!-- Secure HTTPS Protocol
<service name="ModusSuite.MWS.MWSProcessServiceBasic"
behaviorConfiguration ="BasicBehaviour_SSL" >
<host>
  <baseAddresses>
    <add baseAddress="https://localhost:8111/mwsbasic" />
  </baseAddresses>
</host>
<endpoint address="mwsprocess"
```

```
          binding="basicHttpBinding"
          bindingConfiguration="BasicBinding_SSL"
          contract="ModusSuite.MWS.Types.IMWSProcessServiceBasic" />
</service> -->
```

5. Find the following line.

```
<service name="ModusSuite.MWS.MWSProcessServiceRest"
behaviorConfiguration ="RestBehaviour">
```

6. Uncomment the **Insecure HTTP Protocol** part and comment out the **Secure HTTPS Protocol** part as shown in the following example.

   Example

```
<!-- Insecure HTTP Protocol -->
<service name="ModusSuite.MWS.MWSProcessServiceRest"
behaviorConfiguration ="RestBehaviour" >
<host>
  <baseAddresses>
    <add baseAddress ="http://localhost:8011/mwsrest" />
  </baseAddresses>
</host>
<endpoint address="mwsprocess"
          binding="webHttpBinding"
          contract="ModusSuite.MWS.Types.IMWSProcessServiceBasic"
          bindingConfiguration=""
          behaviorConfiguration="WebBehavior"/>
</service>
<!-- Secure HTTPS Protocol
<service name="ModusSuite.MWS.MWSProcessServiceRest"
behaviorConfiguration ="RestBehaviour" >
<host>
  <baseAddresses>
    <add baseAddress ="https://localhost:8111/mwsrest" />
  </baseAddresses>
</host>
<endpoint address="mwsprocess"
          binding="webHttpBinding"
          contract="ModusSuite.MWS.Types.IMWSProcessServiceBasic"
          bindingConfiguration="REST_SSL"
          behaviorConfiguration="WebBehavior"/>
</service>
```

7. Save and close the file.
8. To reserve the URL endpoint for non-administrator users and accounts, complete the following substeps:
   1. Identify the Windows user account that is configured to execute the Windows service

Composer.MWS.

2. Open a **Command Prompt** window with Administrator rights.
3. To reserve the MWS URL, execute the following command replacing `Windows-user-account` with the user account used to execute the Windows service **Composer.MWS**.

```
netsh http add urlacl url=http://+:8011/ user=Windows-user-account
```

9. Restart all Content Composer services.

About Configuring the MWS REST API Communication Channel

Enabling the MWS REST API communication endpoint is required in the following cases.

- Your installation includes the Content Composer Web Client.
- You want to use the MWS REST API from a third-party application.

By default, Content Composer uses HTTPS to encrypt the communication between the Web Client and the MWS REST API endpoint, as well as between third-party applications and the endpoint.

You can also configure these communications to use HTTP.

Note: For security reasons, we recommend the use of HTTPS.

For information on how to configure HTTPS, see Enabling HTTPS for the MWS REST API Communication Channel.

For information on how to configure HTTP, see Enabling HTTP for the MWS REST API Communication Channel.

Enabling HTTPS for the MWS REST API Communication Channel

To enable HTTPS, complete the following steps on the Content Composer server:

Prerequisite

An SSL certificate from a Certificate Authority.

This certificate must have a subject name or subject alternative name that matches the DNS name of your Content Composer server.

1. Import your SSL certificate into the certificate store under **Local Computer > Personal**.
2. Verify that **Certification Path** and **Certificate status** are correct.
3. To reserve the URL endpoint for non-administrator users and accounts, complete the following substeps:
    1. Identify the Windows user account that is configured to execute the Windows service **Composer.MWS**.
    2. Open a **Command Prompt** window with Administrator rights.
    3. To reserve the MWS URL, execute the following command, replacing `Windows-user-account` with the user account used to execute the Windows service

Composer.MWS.

```
netsh http add urlacl url=https://+:9010/ user=Windows-user-account
```

4. From the Content Composer installation directory, open **Composer.MWS.exe.config** with a text editor that supports UTF-8.
5. Find the following line within the `<runtimeservices>` element and uncomment it.

```
<service name="mwswebapi"
assembly="ModusSuite.Runtime.MWSRuntimeService"
type="ModusSuite.Runtime.MwsRuntimeServiceWebApi" />
```

Note: If the line is not present, insert it within the `<runtimeservices>` element.

6. Insert the following line within the `<appSettings>` element if it is not already present.

```
<add key="manVarResultXmlElementType" value="manvar2" />
```

7. To configure the MWS REST API, complete the following substeps:
    1. Find the following line.

```
<mwswebapi ipport="9010" enablessl="true" allowedorigins="*"
alivetimeouthour="100" />
```

    2. Ensure that the value of the `enablessl` attribute is set to `true`.
8. Restart all Content Composer services.

Enabling HTTP for the MWS REST API Communication Channel

To enable HTTP, complete the following steps on the Content Composer server:

Note: For security reasons, we recommend the use of HTTPS.

1. From the Content Composer installation directory, open **Composer.MWS.exe.config** with a text editor that supports UTF-8.
2. Find the following line within the `<runtimeservices>` element and uncomment it.

```
<service name="mwswebapi"
assembly="ModusSuite.Runtime.MWSRuntimeService"
type="ModusSuite.Runtime.MwsRuntimeServiceWebApi" />
```

Note: If the line is not present, insert it within the `<runtimeservices>` element.

3. Insert the following line within the `<appSettings>` element if it is not already present.

```
<add key="manVarResultXmlElementType" value="manvar2" />
```

4. To configure the MWS REST API, complete the following substeps:
    1. Find the following line.

```
<mwswebapi ipport="9010" enablessl="true" allowedorigins="*"
alivetimeouthour="100" />
```

2. Set the value of the `enablessl` attribute to `false`.

```
<mwswebapi ipport="9010" enablessl="false" allowedorigins="*"
alivetimeouthour="100" />
```

5. To reserve the URL endpoint for non-administrator users and accounts, complete the following substeps:
    1. Identify the Windows user account that is configured to execute the Windows service **Composer.MWS**.
    2. Open a **Command Prompt** window with Administrator rights.
    3. To reserve the MWS URL, execute the following command, replacing `Windows-useraccount` with the user account used to execute the Windows service **Composer.MWS**.

```
netsh http add urlacl url=http://+:9010/ user=Windows-
useraccount
```

6. Restart all Content Composer services.

Checking the Availability of the MWS REST API

To check whether the MWS REST API is available, complete one of the following steps in a Web browser.

- If the **HTTPS** communication is enabled, call the following URL, replacing **CoCoServer** with the DNS host name of the Content Composer server that hosts the MWS REST API.

  https://CoCoServer:9010/mws/health
- If the **HTTP** communication is enabled, call the following URL, replacing **CoCoServer** with the DNS host name of the Content Composer server that hosts the MWS REST API.

  http://CoCoServer:9010/mws/health

Result The browser displays the text **Healthy**.

About Configuring Hyland IdP

Content Composer Studio and **Content Composer Angular Web Client** support the usage of the Hyland Identity Provider (IdP).

The following topics describe how to configure Content Composer to use Hyland IdP.

For information on how to configure the **Content Composer Angular Web Client** to use Hyland IdP, see "Configuring Content Composer Web Client to Use Hyland IdP" in the *Content Composer Web Client Installation Guide*.

Prerequisites

Before configuring Content Composer to use IdP, the Hyland IdP server must be prepared.

Request your IdP administrator to create a new IdP client for Content Composer. Provide the administrator with the following information.

Redirect URIs

- **For Content Composer Studio**: http://localhost:4200/auth
- **For Content Composer Web Client**: Host name and IP port of the server that hosts the Web Client files in the following form:

  https://*[WEB-SERVER:IP-Port]*/view/authentication-confirmation

Post Redirect URIs

- **For Content Composer Web Client**: Host name and IP port of the server that hosts the Web Client files in the following form:

  https://*[WEB-SERVER:IP-PORT]*/view/unauthenticated

Supported Grant Types

- Authorization Code with PKCE
- Client Credentials
- Resource owner password grant

Allowed Scopes

- openid
- group
- evolution
- profile.group
- iam.user-catalog.read
- offline_access

Use Client Secrets?

- Yes

Configuring Content Composer IdP Authentication

To configure the Content Composer user rights management for IdP authentication, complete the following steps:

1. Contact your IT department and ask for the user group names that exist in the IdP user store.
2. To identify the role mapping object used by your Content Composer installation, from the **%Composerdir%** directory, open **UserRepository.config** with a text editor that supports UTF-8.
3. Search for the attributes `roleMapper` and `systemOId`.
   - The `roleMapper` attribute contains the used role mapping object.
   - The `systemOId` attribute contains the name of the system in which this role mapping object is stored.
4. In **Content Composer Studio**, switch to the system specified in the `systemOId` attribute.
5. Open the role mapping object specified in the `roleMapper` attribute.
6. In the role mapping object, add a new group for each IdP user group.

   Note: If the IdP user group name contains spaces, replace each space with an underscore.

Example: If the IdP user group name is **GRP - All Employees**, name the new group **GRP_-_ All_Employees**.

7. Assign the required roles to the newly created groups.
8. Save and close the role mapping object.

Modifying the IdP Configuration File

The file **UserRepository_Idp.config** contains the general IdP settings required by Content Composer Studio and backend applications. Complete the following steps on the Content Composer Server and on each computer on which Content Composer Studio is installed.

1. From the **%Composerdir%** directory, open the **UserRepository_Idp.config** file with a text editor that supports UTF-8.
2. Find the following attributes and update their values to the values of your IdP server configuration provided by your IdP administrator.
   - `authority`
   - `redirectUri`: Use the redirectUri for Content Composer Studio.
   - `audience`
   - `idpConfigurationURL`

   Example

```
<?xml version="1.0" encoding="utf-8" ?>
   <userRepository_Idp
     authority = "https://[IDP-WEB-SERVER]/idp"
     redirectUri = "http://localhost:4200/auth"
     audience="https://[IDP-WEB-SERVER]/idp/resources"
     idpConfigurationURL = "https://[IDP-WEB-SERVER]/idp/.well-
known/openid-configuration"
     scopeAuthorizationCode= "openid group iam.user-catalog.read
offline_access"
     scopePasswordGrant = "openid group iam.user-catalog.read
offline_access"
     groupKey = "group"
     userKey = "username"
/>
```

3. Save and close the file.

Testing the IdP Configuration

To verify if Hyland IdP is configured correctly, complete the following steps:

1. From the **%Composerdir%** directory, launch **IdpUserCrypt.exe**.
2. If you are using IdP authentication type **Password Grant**:
   1. Select **Password Grant**.
   2. Enter the appropriate values provided by your IdP administrator in the following fields

and then click **Test login**.

- **IdP User Name**
- **IdP Password**
- **IdP Client Secret**
- **IdP Client ID**

The tool either returns **Login successful** or an error message.

In case of an error message, verify your input and the IdP configuration and then repeat the step.

If you made changes to the **UserRepository_Idp.config** file, restart **IdpUserCrypt.exe**.

3. If you are using IdP authentication type **Client Credentials**:
    1. Select **Client Credentials**.
    2. Enter the appropriate values provided by your IdP administrator in the following fields and then click **Test login**.
        - **IdP Client Secret**
        - **IdP Client ID**

        The tool either returns **Login successful** or an error message.

        In case of an error message, verify your input and the IdP configuration and then repeat the step.

        If you made changes to the **UserRepository_Idp.config** file, restart **IdpUserCrypt.exe**.

Configuring a Content Composer Server Installation to Use Hyland IdP

To configure a Content Composer Server installation to use Hyland IdP authentication, complete the following steps:

1. From the **%Composerdir%** directory, open the **Composer.Core.exe.config** file with a text editor that supports UTF-8.
2. Find the `<configSections>` element and verify that the following line exists within the element.

```
<configSections>
...
  <section name="userRepository_Idp"
type="ModusSuite.Common.SystemFramework.OAuth.IdpConfiguration,
ModusSuite.Common.SystemFramework"/>
...
</configSections>
```

3. Find the line beginning with `<userRepository_Ldap configSource` and verify that the following line exists below this line.

```
<userRepository_Idp configSource="UserRepository_Idp.config" />
```

4. Find the `<behavior name="STSBehaviour">` element and add or uncomment the `<serviceCredentials>` element within the `behavior` element.

   Example

   ```
   <behavior name="STSBehaviour">
     ...
     <serviceCredentials>
       <userNameAuthentication userNamePasswordValidationMode="Custom"
   customUserNamePasswordValidatorType="ModusSuite.Runtime.STS.IdpUserNa
   mePasswordValidator, ModusSuite.Runtime.STSRuntimeService" />
     </serviceCredentials>
     ...
   </behavior>
   ```

5. Save and close the file.
6. From the **%Composerdir%** directory, open the **UserRepository.config** file with a text editor that supports UTF-8.
7. Set the value of the `userStore` attribute to `Idp`.

   Example

   ```
   <userRepository systemOId="cc" roleMapper="Std_Mapping"
   userProfile="Std_Profil" profileReadOption="None" userStore="Idp" />
   ```

8. Save and close the file.
9. To activate the configuration changes, in **Windows Services**, restart the **Composer.Core** service.
10. From the **%Composerdir%** directory, open the **Composer.MWS.exe.config** file with a text editor that supports UTF-8.
11. Find the `<configSections>` element and verify that the following line exists within the element.

    ```
    <configSections>
    ...
      <section name="userRepository_Idp"
    type="ModusSuite.Common.SystemFramework.OAuth.IdpConfiguration,
    ModusSuite.Common.SystemFramework"/>
    ...
    </configSections>
    ```

12. Find the line beginning with `<userRepository_Ldap configSource` and verify that the following line exists below this line.

    ```
    <userRepository_Idp configSource="UserRepository_Idp.config" />
    ```

13. Save and close the file.
14. To activate the configuration changes, in **Windows Services**, restart the **Composer.MWS** service.

Configuring a Content Composer Studio Installation to Use Hyland IdP

To configure a Content Composer Studio installation to use Hyland IdP authentication, complete the following steps:

1. Copy the file **UserRepository_Idp.config** into the Content Composer Studio installation directory.
2. From the **%Composerdir%** directory, open the **Composer.Studio.exe.config** file with a text editor that supports UTF-8.
3. Find the `<configSections>` element and verify that the following line exists within the element.

```
<configSections>
...
  <section name="userRepository_Idp"
type="ModusSuite.Common.SystemFramework.OAuth.IdpConfiguration,
ModusSuite.Common.SystemFramework"/>
...
</configSections>
```

4. Find the line beginning with `<odinSettings configSource=` and add the following line directly below.

```
<userRepository_Idp configSource="UserRepository_Idp.config"/>
```

5. Find the `<appSettings>` element and add the following lines within the element, replacing `[Studio-IDP-Client-Id]` with the IdP client id.

```
<appSettings>
...
  <add key="idpClientId" value="[Studio-IDP-Client-Id]" />
  <add key="credentials" value="Idp" />
...
</appSettings>
```

6. Find the `<runtime>` element and add the `assemblyBinding` element as in the following example.

   Example

```
<runtime>
    <generatePublisherEvidence enabled="false" />
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="Newtonsoft.Json"
publicKeyToken="30ad4fe6b2a6aeed" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-13.0.0.0"
newVersion="13.0.0.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="System.IdentityModel.Tokens.Jwt"
publicKeyToken="31bf3856ad364e35" culture="neutral" />
```

```
        <bindingRedirect oldVersion="0.0.0.0-6.5.1.0"
newVersion="6.5.1.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="Microsoft.IdentityModel.Tokens"
publicKeyToken="31bf3856ad364e35" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-6.5.1.0"
newVersion="6.5.1.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="Microsoft.Extensions.Logging"
publicKeyToken="adb9793829ddae60" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-2.2.0.0"
newVersion="2.1.1.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity
name="Microsoft.Extensions.Logging.Abstractions"
publicKeyToken="adb9793829ddae60" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-2.2.0.0"
newVersion="2.2.0.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="Microsoft.Extensions.Options"
publicKeyToken="adb9793829ddae60" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-2.2.0.0"
newVersion="2.2.0.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity
name="System.Runtime.CompilerServices.Unsafe"
publicKeyToken="b03f5f7f11d50a3a" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-6.0.0.0"
newVersion="6.0.0.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="System.Memory"
publicKeyToken="cc7b13ffcd2ddd51" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-4.0.1.2"
newVersion="4.0.1.2" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="System.Buffers"
publicKeyToken="cc7b13ffcd2ddd51" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-4.0.3.0"
```

```
newVersion="4.0.3.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="System.Text.Encodings.Web"
publicKeyToken="cc7b13ffcd2ddd51" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-6.0.0.0"
newVersion="6.0.0.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="IdentityModel"
publicKeyToken="e7877f4675df049f" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-4.6.0.0"
newVersion="4.6.0.0" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
```

7. Save and close the file.

Content Composer Windows Services and Console Applications with Hyland IdP

For Content Composer Windows services, you can use the authentication types **Password Grant** or **Client Credentials**, with **Password Grant** being the preferred authentication type. Use the authentication type **Client Credentials** only in case no OnBase user store is available in your infrastructure.

Complete the steps in the following topics to configure a Content Composer Windows service or console application to use Hyland IdP authentication.

- Creating the Secrets File
- Configuring Content Composer Windows Services and Console Applications to Use Hyland IdP
- If you are using the authentication type **Client Credentials**, complete the steps in Configuring Content Composer Windows Service Authorization for IdP Authentication Type Client Credentials

About the Secrets File

A Content Composer Windows service or console application uses a specific IdP user account for authentication.

The user credentials and other IdP settings are stored in a so-called *Secrets* file, which is encrypted using the **Windows Data Protection API**.

The Windows Data Protection API uses the windows user account to encrypt and decrypt data.

Therefore, the *Secrets* file can only be decrypted using the same Windows account that was used for encryption.

Creating the Secrets File

To create the **Secrets** file, complete the following steps:

1. Log on to Windows with the user account that is used to execute the Content Composer Windows service or console application.
2. From the **%Composerdir%** directory, launch **IdpUserCrypt.exe**.
3. If you are using authentication type **Password Grant**:
    1. Select **Password Grant**.
    2. Enter the appropriate values provided by your IdP administrator in the following fields.
        - **IdP User Name**
        - **IdP Password**
        - **IdP Client Secret**
        - **IdP Client ID**: This value is only needed to test the logon and is not saved in the Secrets file.
4. If you are using authentication type **Client Credentials**:
    1. Select **Client Credentials**.
    2. Enter the appropriate values provided by your IdP administrator in the following fields.
        - **IdP Client Secret**
        - **IdP Client ID**
5. In the **Filename** field, enter the name of the **Secrets** file.
6. Optional. Click **Test login**.

    The tool either returns **Login successful** or an error message.

    In case of an error message, verify your input and the IdP configuration and then repeat the previous step.

    If you made changes to the **UserRepository_Idp.config** file, restart **IdpUserCrypt.exe**.
7. Click **Encrypt to file** to create an encrypted file that contains the IdP parameters.

Configuring Content Composer Windows Services and Console Applications to Use Hyland IdP

To configure user-defined Content Composer Windows services to use Hyland IdP authentication, complete the following steps:

Note: These instructions do not apply to the services Composer.Core, Composer.MWS, Composer.OWS, and Composer.XWS.

1. From the **%Composerdir%** directory, open the respective CONFIG file with a text editor that supports UTF-8.
2. Find the `<configSections>` element and verify that the following line exists within the element.

    ```
    <configSections>
    ```

```
...
  <section name="userRepository_Idp"
type="ModusSuite.Common.SystemFramework.OAuth.IdpConfiguration,
ModusSuite.Common.SystemFramework"/>
...
</configSections>
```

3. Find the line beginning with `<odinSettings configSource="` and verify that the following line exists below.

```
<userRepository_Idp configSource = "UserRepository_Idp.config"/>
```

4. Find the `<appSettings>` element and add the following lines within the element, replacing `Service-IDP-Client-Id` with the IdP client ID and `idpSecretsFile` with the name of the file you created in Creating the Secrets File

   Important: Specify the full file path of the **Secrets** file. Environment variables are supported.

```
<appSettings>
...
  <add key="idpClientId" value="Service-IDP-Client-Id" />
  <add key="credentials" value="IdpPasswordGrant" />
  <add key="idpSecretsFile" value="%ComposerDir%\idpSecret_
JohnMallory.txt" />
...
</appSettings>
```

5. If you are using authentication type **Client Credentials**, modify the value of the key `credentials` to `IdpClientCredentials`.

```
<appSettings>
...
  <add key="idpClientId" value="Service-IDP-Client-Id" />
  <add key="credentials" value="IdpClientCredentials" />
  <add key="idpSecretsFile" value="%ComposerDir%\idpSecret_
JohnMallory.txt" />
...
</appSettings>
```

6. Find the `<runtime>` element and add the `assemblyBinding` element as follows.

```
<runtime>
   <generatePublisherEvidence enabled="false" />
   <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
         <assemblyIdentity
name="Microsoft.Extensions.Logging.Abstractions"
publicKeyToken="adb9793829ddae60" culture="neutral" />
         <bindingRedirect oldVersion="0.0.0.0-2.2.0.0"
```

```
newVersion="2.2.0.0" />
      </dependentAssembly>
      <dependentAssembly>
         <assemblyIdentity name="Microsoft.Extensions.Options"
publicKeyToken="adb9793829ddae60" culture="neutral" />
         <bindingRedirect oldVersion="0.0.0.0-2.2.0.0"
newVersion="2.2.0.0" />
      </dependentAssembly>
      <dependentAssembly>
         <assemblyIdentity
name="System.Runtime.CompilerServices.Unsafe"
publicKeyToken="b03f5f7f11d50a3a" culture="neutral" />
         <bindingRedirect oldVersion="0.0.0.0-6.0.0.0"
newVersion="6.0.0.0" />
      </dependentAssembly>
      <dependentAssembly>
         <assemblyIdentity name="System.Memory"
publicKeyToken="cc7b13ffcd2ddd51" culture="neutral" />
         <bindingRedirect oldVersion="0.0.0.0-4.0.1.2"
newVersion="4.0.1.2" />
      </dependentAssembly>
      <dependentAssembly>
         <assemblyIdentity name="System.Buffers"
publicKeyToken="cc7b13ffcd2ddd51" culture="neutral" />
         <bindingRedirect oldVersion="0.0.0.0-4.0.3.0"
newVersion="4.0.3.0" />
      </dependentAssembly>
      <dependentAssembly>
         <assemblyIdentity name="System.Text.Encodings.Web"
publicKeyToken="cc7b13ffcd2ddd51" culture="neutral" />
         <bindingRedirect oldVersion="0.0.0.0-6.0.0.0"
newVersion="6.0.0.0" />
      </dependentAssembly>
      <dependentAssembly>
         <assemblyIdentity name="IdentityModel"
publicKeyToken="e7877f4675df049f" culture="neutral" />
         <bindingRedirect oldVersion="0.0.0.0-4.6.0.0"
newVersion="4.6.0.0" />
      </dependentAssembly>
      <dependentAssembly>
         <assemblyIdentity name="Newtonsoft.Json"
publicKeyToken="30ad4fe6b2a6aeed" culture="neutral" />
         <bindingRedirect oldVersion="0.0.0.0-13.0.0.0"
newVersion="13.0.0.0" />
      </dependentAssembly>
```

```
    </assemblyBinding>
  </runtime>
```

7. Save and close the file.
8. Repeat the previous steps for each user-defined Content Composer Windows service and console application.
9. In **Windows Services**, change the **Log on as** property from **Local System account** to **This account** using the credentials of the user used to create the **Secrets** file.

Configuring Content Composer Windows Service Authorization for IdP Authentication Type *Client Credentials*

Use the authentication type **Client Credentials** only in case no OnBase user store is available in your infrastructure. We highly recommend using the authentication type **Password Grant**.

The **Client Credentials** authentication type uses only a `Client ID` and a `Secret` value for authentication. There is no user name with password and therefore no user context.

Content Composer Windows Service uses the `Client ID` as the Content Composer username and group name.

Therefore, you must add the `Client ID` as a group to the Content Composer user rights management to be able grant rights.

To configure the Content Composer user rights management for **Client Credential** authentication, complete the following steps:

1. To identify the **IdP Client ID** your Content Composer service is using, from the **%Composerdir%** directory, open the CONFIG file of the respective service with a text editor that supports UTF-8.
2. Find the `idpClientId` key within the `<appSettings>` element. Its value specifies the ID in the service application configuration file.
3. To identify the role mapping object used by your Content Composer installation, from the **%Composerdir%** directory, open **UserRepository.config** with a text editor that supports UTF-8.
4. Search for the attributes `roleMapper` and `systemOId`.
   - The `roleMapper` attribute contains the used role mapping object.
   - The `systemOId` attribute contains the name of the system in which this role mapping object is stored.
5. In **Content Composer Studio**, switch to the system specified in the `systemOId` attribute.
6. Open the role mapping object specified in the `roleMapper` attribute.
7. In the role mapping object, add a new group for each **IdP Client ID** you are using.

   Note: If the **IdP Client ID** name contains spaces, replace each space with an underscore.

   Example: If the IdP Client ID name is **Composer Service**, name the new group **Composer_ Service**.
8. Assign the required roles to the newly created groups.
9. Save and close the role mapping object.

About Configuring Content Composer Web Client to Use Hyland IdP

For information on how to configure a Content Composer Web Client to use Hyland IdP authentication, see "Configuring Content Composer Web Client to Use Hyland IdP" in the *Content Composer Web Client Installation Guide*.

Installing Content Composer Studio only

To install and customize Content Composer Studio only, without Client or Server components, complete the following steps:

Prerequisite

A Content Composer Server installation is required.

1. Install Content Composer Studio. See Installing Content Composer Studio for more information.
2. Import the RSA encryption keys. See Importing the RSA Encryption Keys for more information.
3. Grant access rights to the RSA container. See Granting Access Rights to the RSA Container for more information.
4. Configure Content Composer Studio. See Configuring Content Composer Studio for more information.

Installing Content Composer Studio

To install Content Composer Studio, complete the following steps:

1. In **Windows File Explorer**, right-click **Content Composer Setup 23.1.0 for Windows (64bit).exe**, and then click **Properties**.
2. If the **Unblock** check box is displayed at the bottom right of the **Content Composer Setup 23.1.0 for Windows (64bit).exe Properties**, dialog box, select **Unblock**.

   Note: If the **Unblock** check box is not displayed, the file is already unblocked.
3. Click **OK**.
4. Double-click the **Content Composer Setup 23.1.0 for Windows (64bit).exe** file.
5. Optional. In the dialog box, click **Install** if **Microsoft Visual Studio Tools for Office Runtime** tool is not installed on your system,
6. In the **Welcome to the Content Composer Setup Wizard** page, click **Next**.
7. In the **License Agreement** page, review the terms in the License Agreement, scroll to the end of the agreement, click **I accept the terms in the license agreement**, and click **Next**.
8. In the **Destination Folder** page, accept the default directory, and click **Next** or click **Change** to select an alternate directory.
9. In the **Setup Type** page, click **Custom**.
10. In the **Custom Setup** page, select the **Content Composer Studio** feature and clear all other features.
11. In the **Ready to Install the Program** page, click **Install**.
12. In the **Content Composer Installation Completed** page, click **Finish**.

Importing the RSA Encryption Keys

If you are performing an additional installation of Content Composer components, you need to import the encryption keys created with your first Content Composer server installation. Complete the following steps:

1. From your secure backup store, copy the following files to the **%Composerdir%** directory:
   - encryption_keys.config
   - CoCoEncryptionRSAKey.xml
2. Open a command prompt window as an administrator and navigate to the **%Composerdir%** directory.
3. To import the encryption key, execute the following command:

   ```
   Composer.EncryptionTool importkey
   ```

4. For security reasons, delete `CoCoEncryptionRSAKey.xml` from the **%Composerdir%** directory after importing.

Granting Access Rights to the RSA Container

To grant access rights to the RSA container to every Windows account that runs a Content Composer service or Studio, complete the following steps:

1. Open a command prompt window as an administrator and navigate to the **%windir%\Microsoft.NET\Framework64\v4.0.30319** directory.
2. Execute the following command, which grants access rights to the user `CindySmith`, who is a member of the domain `OnBase`, replacing domain and user name with the required values:

   ```
   aspnet_regiis -pa coco "onbase\cindysmith"
   ```

3. Repeat the previous step for each Windows account running a Content Composer service or Studio.

Configuring Content Composer Studio

To configure your Content Composer installation, complete the following steps:

1. In the installation directory, double-click the **Composer.SetupAssistant.exe** file.
2. On the **Content Composer Setup** tab, click **Next**.
3. In the **Content Composer Software Configuration** page, select the location of the Content Composer services, in the **Computer name** box, type the computer name.
4. In the **Content Composer System Selection** page, select the base system and click **Next**.
5. In the **Content Composer Microsoft Word Template** page, select the required Normal.dotm file and click **Next**.
6. In the **Close the Content Composer Assistant** page, select one of the following options and click **OK**.
   - Restart the assistant
   - Exit

Confirming Composer Word Add-In Installation

The Content Composer Setup delivers the Composer.WordAddIn. This add-in for Microsoft Word enables the user to insert text block variables directly into a Word document.

Most combinations of Microsoft Windows and Microsoft Word installations require confirmation of the Word AddIn installation. To confirm the installation of Composer.WordAddIn, complete the following steps:

1. Start Microsoft Word.
2. If the Microsoft Office Customization Installer dialog box appears, click **Install**.
3. On the toolbar, click **File > Options**.
4. In the left pane, click **Add-ins**. If **Composer.WordAddIn** is already installed on your system, **Content Composer Word AddIn** exists under **Active Application Add-ins**.
5. If there are issues in installing the Content Composer Word Add-In, complete the steps described in the Content Composer Word Add-In section.

Starting the Core Service

To start the Core service, complete the following step:

- Open **Windows Services** and start the **Composer.Core** service.

Launching Content Composer Studio

Ensure that the Windows service **Composer.Core** is running on the Composer Server installed system. To start the Studio, complete the following step:

1. Ensure that the **Composer.Core** service is running on the Content Composer server.
2. Launch **Content Composer Studio**.
3. When launching Content Composer Studio for the first time, important controls, such as the **Navigator**, are not visible. To display these controls, complete the following steps:
    - On the **Studio** tab, click the **Navigator** button.
    - On the **Studio** tab, click the **Object Inspector** button.
    - On the **Studio** tab, click the **Output** button.
    - On the **Studio** tab, click the **Toolbox** button.

Result You have now installed and customized Content Composer Studio with German and English as user languages. To install additional language packs, see the User Language section. Follow the instructions with all sections that denote Content Composer Studio and Client in the heading.

Installing Content Composer Client only

To install and customize Content Composer Client only, without Studio or Server components, complete the steps in the following sections.

1. Installing Content Composer Client
2. Configuring Content Composer Client

Installing Content Composer Client

To install the Content Composer Client only, without Studio and Server components, complete the following steps:

1. In **Windows File Explorer**, right-click **Content Composer Setup 23.1.0 for Windows (64bit).exe**, and then click **Properties**.
2. If the **Unblock** check box is displayed at the bottom right of the **Content Composer Setup 23.1.0 for Windows (64bit).exe**, dialog box, select **Unblock**.

   Note: If the **Unblock** check box is not displayed, the file is already unblocked.
3. Click **OK**.
4. Double-click the **Content Composer Setup 23.1.0 for Windows (64bit).exe** file.
5. Optional. In the dialog box, click **Install** if Microsoft Visual Studio Tools for Office Runtime tool is not installed on your system.
6. In the **Welcome to the Content Composer Setup Wizard** page, click **Next**.
7. In the **License Agreement** page, review the terms in the License Agreement, scroll to the end of the agreement, click **I accept the terms in the license agreement**, and click **Next**.
8. In the **Destination Folder** page, accept the default directory, and click **Next** or click **Change** to select an alternate directory.
9. In the **Setup Type** page, click **Custom**.
10. In the **Custom Setup** page, select the **Content Composer Client** feature and clear all other features.
11. In the **Ready to Install the Program** page, click **Install**.
12. In the **Content Composer Installation Completed** page, click **Finish**.

Configuring Content Composer Client

To configure your Content Composer Client installation, complete the following steps:

1. In the installation directory, double-click the **Composer.SetupAssistant.exe** file.
2. On the **Content Composer Setup** tab, click **Next**.
3. In the **Content Composer Software Configuration** page, select the location of the Content Composer services, in the **Computer name** box, type the computer name.
4. In the **Content Composer System Selection** page, select the base system and click **Next**.
5. In the **Close the Content Composer Assistant** page, select one of the following options and click **OK**.
   - Restart the assistant
   - Exit

User Language

This section includes the installation processes you require to follow to install the user languages. See the following sections for additional information.

- Installation Packages for Additional Languages (Content Composer Studio and Client)
- Installing a Language Package (Content Composer Studio and Client)
- Setting the Language for the Odin Views (Content Composer Server)
- Running Content Composer in Another Language
- Unattended Installation and Uninstallation of a Language Package

Installation Packages for Additional Languages (Content Composer Studio and Client)

Content Composer offers installation packages that allow you to install additional languages on your system. The language packs for German and English (US) are delivered and installed with the product setup.

Additionally, Polish and Dutch language packages are available upon request.

Installing a Language Package (Content Composer Studio and Client)

To install a language package on your system, complete the following steps:

Prerequisite

A Content Composer Studio or Client installation

1. To obtain the language package, contact the Hyland Software Technical Support group.
   For a list of Technical Support phone numbers, go to hyland.com/pswtscontact.
2. Save the file locally, so that you can access it during the update procedures.
3. Double-click the **Content Composer [*Language*] Language Pack 64 bit.exe** file.
4. In the **Welcome to the Content Composer [*Language*] Language Pack Wizard** page, click **Next**.
5. In the **License Agreement** page, review the terms in the License Agreement, scroll to the end of the agreement, click **I accept the terms in the license agreement**, and click **Next**.
6. In the **Destination Folder** page, click **Next**.
7. In the **Ready to Install the Program** page, click **Install**.
8. In the **Content Composer Installation Completed** page, click **Finish**.

Setting the Language for the Odin Views (Content Composer Server)

To set the language for the Odin views, complete the following steps:

1. Open your database management system.
2. Open the ODIN_SETTING table.
3. Change the OD_ISO_LANG_CODE field to one of the following language codes. The language codes are case-sensitive.

Language Code	Language
en-US	English (US)
Fr	French
Nl	Dutch
Es	Spanish
pt-BR	Portuguese (BR)

Running Content Composer in Another Language

To run Content Composer Studio or Client in a language other than the one defined in the Windows system settings, complete the following steps:

1. Navigate to the installation directory and then to one of the following subdirectories.
   - **de** for German
   - **en** for English
   - **fr** for French
   - **nl** for Dutch
   - **pl** for Polish
2. To start Content Composer Studio, double-click **Composer.Studio.cmd**.
3. To start Content Composer Client, double-click **Composer.Client.cmd**.

Unattended Installation and Uninstallation of a Language Package

Review the following information about installing or uninstalling a language package in silent mode.

- Specify the Content Composer installation directory with the parameter `INSTALLDIR`. The setup installs the language pack in a subdirectory of the Content Composer installation directory.
- Specify command line options, which require a parameter, without spaces between the option and its parameter.

  Valid example: `Setup.exe /v"/qn"`

  Invalid example: `Setup.exe /v "/qn"`
- Quotation marks around the parameter of an option are only required if the parameter contains spaces.

  If a path within a parameter contains spaces, enclose the path in quotation marks as well.

  Example `Setup.exe /v"INSTALLDIR=\"c:\My Files\""`

Installing a Language Package Unattended

To perform an unattended installation of a Content Composer language package, complete the following steps:

1. Navigate to the directory that contains the file **ContentComposer_Unattended-Installation_CMDSamples.zip** and unzip the file.
2. Open **Install_CoCo_LanguagePack.cmd** with a text editor.
3. Replace `[EXE-Setup-File]` with the name of the required EXE file.
    - `Content Composer Polish Language Pack 64 Bit.exe`
    - `Content Composer Dutch Language Pack 64 Bit.exe`
4. Save and close the file.
5. Execute the CMD file.

Uninstalling a Language Package Unattended

To perform an unattended uninstallation of a Content Composer language package, complete the following steps:

1. Navigate to the directory that contains the file **ContentComposer_Unattended-Installation_CMDSamples.zip** and unzip the file.
2. Open **UnInstall_CoCo_LanguagePack.cmd** with a text editor.
3. Replace `[EXE-Setup-File]` with the name of the required EXE file.
    - `Content Composer Polish Language Pack 64 Bit.exe`
    - `Content Composer Dutch Language Pack 64 Bit.exe`
4. Save and close the file.
5. Execute the CMD file.

Unattended Installation and Uninstallation of Content Composer

The Content Composer installation package provides predefined CMD files that you can use for an unattended installation or uninstallation.

CMD File	Description
Install_CoCo_Complete.cmd	Installs all Content Composer components.
Install_CoCo_Studio_64bit.cmd	Install the 64bit Studio version only.
Install_CoCo_Studio_Client_64bit.cmd	Installs the 64bit Studio and Client version.
Uninstall_CoCo.cmd	Uninstalls all Content Composer components.

Parameters

The following table lists the parameters used for the silent installation and uninstallation of Content Composer.

<table>
<tr><th>Parameter</th><th colspan="2">Description</th></tr>
<tr><td>/s</td><td colspan="2">Runs the installation or uninstallation in silent mode.</td></tr>
<tr><td>/x</td><td colspan="2">Uninstalls the product.</td></tr>
<tr><td rowspan="5">/v”Parameter”</td><td colspan="2">Passes the Windows installer (MSI) parameters to the Windows installer.
Notes
• Specify command line options, which require a parameter, without spaces between the option and its parameter.
Valid example: <code>Setup.exe /v"/qn"</code>
Invalid example: <code>Setup.exe /v "/qn"</code>
• Quotation marks around the parameter of an option are only required if the parameter contains spaces.
If a path within a parameter contains spaces, enclose the path in quotation marks as well.
Example<code>Setup.exe v"/l*v \"C:\Composer Log File.txt\""</code></td></tr>
<tr><td colspan="2">You can pass the following parameters to MSI.</td></tr>
<tr><td>/qn</td><td>Performs a silent installation or uninstallation.</td></tr>
<tr><td>/l*v logfilepath</td><td>Sets the path and filename of install log file.</td></tr>
<tr><td>ADDLOCAL= Component</td><td>Specifies the Content Composer components to install.
• Client. Installs Content Composer Client.
• Studio64. Installs Content Composer Studio.
• Server64. Installs Content Composer Server.
Note Use a comma to separate multiple instructions.</td></tr>
</table>

Installing Content Composer Unattended

To install Content Composer unattended, complete the following steps:

1. Navigate to the directory that contains the file **ContentComposer_Unattended-Installation_CMDSamples.zip** and unzip the file.
2. Open the required file with a text editor.
   - Install_CoCo_Complete.cmd
   - Install_CoCo_Studio_64bit.cmd
   - Install_CoCo_Studio_Client_64bit.cmd
3. Replace `[EXE-Setup-File]` with `Content Composer Setup 23.1.0 for Windows (64bit).exe`.
4. Save and close the file.
5. Execute the CMD file.

Uninstalling Content Composer Unattended

To install Content Composer unattended, complete the following steps:

1. Navigate to the directory that contains the file **ContentComposer_Unattended-Installation_CMDSamples.zip** and unzip the file.
2. Open **Uninstall_CoCo.cmd** with a text editor.
3. Replace `[EXE-Setup-File]` with `Content Composer Setup 23.1.0 for Windows (64bit).exe`.
4. Save and close the file.
5. Execute the CMD file.
6. If the **Content Composer Setup 23.1.0 for Windows (64bit).exe** file is not available, complete the following steps to uninstall the complete product.
   1. Open a **Command Prompt** window with Administrator rights.
   2. Execute the following command.

      ```
      msiexec /x {C6147AED-4AC4-44EB-A925-BC6D57225D4F}
      ```

Advanced Installation

You can also configure the Content Composer installation without using the Composer Setup Assistant. This process is intended for experienced users.

Odin or MWS Database Setup

About the setup (Content Composer Server)

The Odin or MWS setup installs Odin and MWS on a database.

For information about the supported databases, see the *Content Composer Technical Specifications*.

Note: You must run this application on the computer on which the Composer Server services are run. Run the Odin/MWS setup from a user account that has write permissions for the Content Composer installation directory.

Prerequisites

License. If you do not have a valid license, send an email to Hyland Support and request a license file for your environment. Copy the license file to the Content Composer installation directory and ensure that the file name is Composer.lic in your database. Verify the following information based on your operating system.

MS SQL Server

- An empty database for the Odin/MWS database schemas.
- An empty database for the repository DB schema.
- The SQL Servers authentication mode must be active.
- The database roles **composer_full** and **composer_readonly**.

  To create these roles, complete the steps in Creating the Database Users in SQL Server.
- At least one user who meets the following criteria.
  - The user can access the Content Composer databases.
  - The user is a member of the **composer_full** role.

Oracle

- An Oracle user for the database in which the Odin/MWS and the repository database schema is to be installed.

  To create these users, complete the steps in Creating the Oracle Users

To install Odin and MWS on a database, you can either use the Database Setup wizard or you can manually execute the required SQL scripts.

Setting up the Database

To install Odin and MWS on a database using the Database Setup wizard, complete the following steps:

1. In the **Content Composer installation** directory, start the **Composer.Database.Setup.exe**.
2. On the **Composer Database Setup** page, verify the installation prerequisites. Each step of the list must show OK before you can proceed to the next page.
3. Optional. Resolve all indicated errors and click **Test**.
4. Click **Next**.
5. On the **Select Procedure** page, select **Create Odin/MWS Schema** and click **Next**.
6. On the **Database Connection** page, select one of the following database products and complete the appropriate procedure.

Option	Description
MS SQL Server - Odin/MWS database	On the **SQL Server Connection** page, complete the following steps: 1. In the **Enter or select server name** box, type the name of the SQL Server instance. 2. In the **Server logon** box, complete one of the following substeps: • Select the **Use SQL Server Authentication** check box. In the **User** box, type the name of the database user you want to use to create the database schema and in the **Password** box, type the password of the database user **Note**: Use the database user created in Create the database users in SQL Server that has the **composer_full** role. • Select the **Use Windows authentication** check box. 3. In the **Database** box, type the name of the SQL Server database to which you want to install the Odin and MWS database schema. 4. Click **Test Connection** and click **OK**.
Oracle - Odin/MWS database	In the **Oracle connection Properties** page, complete the following steps: 1. In the **Enter server name** box, type the name of the Oracle server instance.

Option	Description
	2. In the **User** box, type the name of the database user you want to use to create the Odin/MWS database schema. **Note**: Use the database user created by the `Ora_Odin_CreateUser.sql` script because the user must have specific permissions. 3. In the **Password** box, type the password of the database user. 4. Click **Test Connection** and click **OK**.

Oracle - Odin/MWS database

7. In the **Select Odin or MWS Option** page, select one of the following options.
   - Create Odin Schema.
   - Create MWS Schema.
   - Create Odin and MWS Schema.
8. Click **Next**. Complete the following substeps:
   1. In the **Installation** page, click **Start installation**.
   2. When the installation is completed, click **OK**.

Installation of Odin or MWS Manually on an Oracle Database (Composer Server)

The following section includes the information on the manual installation of Odin or MWS manually on an Oracle database.

Roles	System permissions
Connect	Create table
Resource	• Create view • Create procedure • Create sequence

You can create a corresponding Oracle user by running the `Ora_Odin_CreateUser.sql` script. The script can generate the following error messages.

- ORA-00942. Table or View does not exist.
- ORA-02289. Sequence does not exist.

You can ignore these messages. They occur because the SQL scripts contain DROP instructions for objects that do not exist in the database.

Database Schema

The subdirectory Admin\Oracle contains the following files. They are used to create the Odin and MWS schemes.

- `Ora_Odin5.sql`
- `Ora_Mws5.sql`

Installation of Odin or MWS Manually on a MS SQL Server Database (Composer Server)

The following section includes the information on the manual installation of Odin or MWS manually on an MS SQL Server database.

Database Schema

The subdirectory *Admin\MS-SQL* contains the following files. They are used to create the Odin and MWS schemas.

- MSSql_Odin5.sql
- MSSql_MWS.sql

Notes

- Run both scripts using an account which is a member of the **composer_full** role.
- You may have to create SQL Server databases to run the scripts.
- You can provide any relevant name to the databases. You require the names of these databases at a later step in the configuration process.

Database User

To create the required database roles and users, complete the steps in Create the database roles and users in SQL Server.

Repository Setup (Content Composer Server)

The repository setup installs the first Content Composer system on a database.

You must run this application on the computer on which you want to run the Content Composer server services. Run the repository setup from a user account that has write permissions for the Content Composer installation directory.

Note: Before running the repository setup, ensure that you possess a license and one of the following databases, which include MS SQL Server or Oracle.

License

If you do not have a valid license, send an email to Hyland Support and request the license for your environment.

Copy the license file to the Content Composer installation directory and ensure that the file name is **Composer.lic**.

Oracle

- Run the `Ora_Rep_CreateUser.sql` script to create an Oracle user for the database in which the Content Composer repository database schema is to be installed. The script is located in the subdirectory *Admin\Oracle*.
- You must have “SYSDBA” permissions to run the script.
- An Oracle user for the database in which the Content Composer repository database schema is to be installed.

  To create these users, complete the steps in Creating the Oracle Users

Installing the Repository (Content Composer Server)

To install the Content Composer repository, complete the following steps:

1. In the **Content Composer** installation directory, start **Composer.Database.Setup.exe**.
2. In the **Content Composer Database Setup** page, verify the Content Composer Repository installation prerequisites. Each step of the list must show **OK** before you can proceed to the next page.
3. In case of errors, resolve all indicated errors and click**Test**.
4. Click **Next**.
5. In the **Select procedure** page, select **Create Repository Schema**.
6. Click **Next**.
7. In the **Database connection** page, select one of the following database products and complete the appropriate procedure.

Option	Description
MS SQL Server - Repository database	On the **SQL Server Connection** page, complete the following steps: 1. In the **Enter or select server name** box, type the name of the SQL Server instance. 2. In the **Server logon** box, complete only one of the following substeps: • Select the **Use SQL Server Authentication** check box. In the **User** box, type the name of the database user you

Option	Description
	want to use to create the repository database schema and in the **Password** box, type the password of the database user. **Note**: Use the database user created in Create the database users in SQL Server that has the **composer_full** role. • Select the **Use Windows authentication** check box. 3. In the **Database** box, type the name of the SQL Server database to which you want to install the repository database schema. 4. Click **Test Connection**. 5. Click **OK** and then click **Next**.
Oracle - Repository database	On the **Oracle Connection Properties** page, complete the following substeps: 1. In the **Enter server name** box, type the name of the Oracle server instance. 2. In the **User** box, type the name of the database user you want to use to create the repository database schema. **Note**: Use the database user that was created by the `Ora_Rep_CreateUser.sql` script because the user must have specific permissions. 3. In the **Password** box, type the password of the database user. 4. Click **Test Connection**. 5. Click **OK** and click **Next**.

8. In the **Creating the System** page, complete the following substeps:
    1. In the **System Name** box, type the name of the Content Composer system you want to create.
    2. In the **SystemObjectID** box, type the ID of the Content Composer system you want to create.

       Note: Enter a unique ID for the new system and write it down. You will need it later during the configuration.
    3. Optional. In the **Database Alias** box, change the predefined value.
9. Click **Next** and in the **Installation** page, click **Start installation**.
10. When the installation is complete, click **OK**.

Modification of the Server Configuration

After the repository setup completes, you must enter the System Object ID you entered during the repository setup in a number of configuration files.

Modify UserRepository.config (Content Composer Server)

To modify the **UserRepository.config** file, complete the following steps:

1. Under the installation directory, start **Composer.ConfigurationEditor**.
2. On the **File** menu, click **Open Configuration**.
3. In the **Open Configuration File** dialog box, select the **UserRepository.config** file and click Open.
4. In the upper pane, in the **Value** column, double-click the **System Object ID** value and enter the system object ID you entered during setup.
5. To save your changes, under the **File** menu, click **Save Configuration**.

Modify Composer.MWS.exe.config (Content Composer Server)

To modify the **Composer.MWS.exe.config** file, complete the following steps:

1. Under the installation directory, start **Composer.ConfigurationEditor**.
2. On the **File** menu, click **Open Configuration**.
3. In the **Open Configuration File** dialog box, select the **composer.MWS.exe.config** file and click **Open**.
4. In the upper pane, in the **Value** column, double-click the **System Object ID** value and enter the system object ID you entered during setup.
5. To save your changes, under the **File** menu, click **Save Configuration**.

Specify Normal.dotm (Content Composer Server)

You must copy Normal.dotm to the installation directory that you specified in the Destination Folder page during the Installing Content Composer procedure.

Modify Composer.Client.exe.config (Content Composer Client)

To modify the **Composer.Client.exe.config** file, complete the following steps:

1. Under the installation directory, start **Composer.ConfigurationEditor**.
2. On the **File** menu, click **Open Configuration**.
3. In the **Open Configuration File** dialog box, select the **Content Composer.Client.exe.config** file and click **Open**.
4. In the upper pane, in the **Value** column, double-click the **Odin View System Object ID** value and enter the system object ID you entered during setup.
5. Optional. You must change the corresponding URL based on whether a service is running on a different computer. Replace the local IP address or localhost with the IP address or name of the server on which the service is run. You must change the URLs localhost entry to the specific server host name or IP address for the following entries.

- Data Provider Service
- License Service
- Login Service
- Composer User Service
- Composer Web Service
- Composer Web Service Repository
- Odin Web Service
- Security Token Service

> **Example**
>
> If the Content Composer Server service is on the ComposerServer host, you must change the Security Token Services URL from http://localhost:8000/sts to http://ComposerServer:8000/sts.

To change the URL, perform one of the following actions.

- Change the URLs for all services.
- Change a specific URLs of a service.

To change the URLs for all services, complete the following substeps:

1. On theTools menu, click Adapt all Service URLs.
2. In the Adapt Service URLs dialog box, type the URL and click OK.
3. To save your changes, on the File menu, click Save Configuration.

To change a specific URLs of a service, complete the following substeps:

1. In the Value column, double-click the value you want to change.
2. Type the URL.
3. To save your changes, on the File menu, click Save Configuration.

Start the Core service (Content Composer Studio)

To start the Content Composer Core service, complete the following steps:

1. Open Windows Services.
2. Start the **Composer.Core** service.

Launching Content Composer Studio

After starting the service, complete the following steps to launch Content Composer Studio.

1. Navigate to **Start > All Programs > Start > All Programs > Content Composer > Content Composer Studio**.

   Note: If you start Content Composer Studio for the first time, important controls, such as the Navigator, are not visible.

2. To display these controls, complete the following substeps:

   1. On the **Studio** tab, click the **Navigator** button.

2. On the **Studio** tab, click the **Object Inspector** button.
3. On the **Studio** tab, click the **Output** button.
4. On the **Studio** tab, click the **Toolbox** button.

Entering Connection Data for the Odin or MWS Database in the Configuration (Content Composer Server)

In Content Composer Studio, complete the following steps:

1. In the top left, click the **Select System** button.
2. In the **Select System** dialog box, select the system and click **OK**.
3. In the **Administration** tab, click the **DB Alias Administration** button. To configure the MWS or Odin database connection, complete the following substeps:
    1. In the **DB Alias Administration** pane, double-click **MWS** or **Odin**.
    2. In the **Data Link Properties** dialog box, complete the following substeps:
        1. Under **Data Provider**, select **Native SQL** for MS SQL Server or **Microsoft Oracle** for an Oracle database.
        2. In the **SQL Template** field, click **Standard Value**. The default value is inserted.
        3. In the **SQL Field Template** field, click the Standard Value. The default value is inserted.
        4. Click the **Ellipsis** button to the right of the **Connection String** box.
        5. In the **Connection Properties** dialog box, enter the connection information for the DB user previously created for the respective schema using the `Ora_Odin_CreateUser.sql` script or the `MSSql_CreateUser.sql` script.

           Note: For SQL Server, use the database user created in Create the database roles and users in SQL Server that has the **composer_full** role.
        6. Ensure that you select the **Allow saving password** option.
        7. Click **OK**.
    3. Repeat the previous steps for Odin or MWS DB alias and save your changes.

Starting the Services

To start the Content Composer services, complete the following steps:

1. Open Windows services.
2. Start the **Composer.Core** service if it is not running.
3. Start the **Composer.MWS** service.
4. Start the **Composer.OWS** service.

Troubleshooting

Content Composer Word Add-In

If the Content Composer Word Add-In is not successfully installed using the setup, execute the following steps:

1. Close all running instances of **Microsoft Word** and **Content Composer Studio**.
2. Navigate to **Windows > Control Panel > Programs and features**.
3. In the list of installed programs, search for the entry **Composer.WordAddIn**, and uninstall the program.
4. Open **Windows Explorer** and navigate to your **Content Composer Studio** installation directory.
5. Double-click the **Composer.WordAddIn.vsto** file. This installs the Content Composer Word Add-In.
6. Start MS Word. To check if the **Content Composer Word Add-In** is activated, perform the following substeps:
    1. On the **File** menu, select **Word options**.
    2. In the left pane, select **Add-Ins**.
    3. Under **Active Application Add-ins**, search for **Content Composer Add-In**.

        If it is available, Content Composer Word Add-In is active in your system.
7. If **Content Composer Add-In** is listed under **Inactive Application Add-ins**, execute the following substeps:
    1. Select **Content Composer Add-In**.
    2. Click **Go**. In the dialog box, select the **Content Composer Add-In** box and click **OK**. The Content Composer Word Add-In is now active on your system.
    3. If an error message is displayed, click **Details**, take a screenshot of the error message, and send it to the Support services.

RSA

Issue: An error message similar to the following one is displayed in the Content Composer Studio Output pane:

```
... Error message from the provider. The RSA key container could not be
opened. ([File-Path]\encryption_keys.config line 2)
```

Solution: Perform the steps in the following topics:

- Importing the RSA Encryption Keys
- Granting Access Rights to the RSA Container

Deleting an RSA Container

To delete an RSA key that is no longer needed:

1. Open a command prompt window as an administrator and navigate to the

%windir%\Microsoft.NET\Framework64\v4.0.30319 directory.

2. Execute the following command:

```
aspnet_regiis -pz coco
```

If the RSA key has been successfully deleted, the following output is displayed:

```
Deleting RSA Key container...
Succeeded!
```