# Content Composer

## Konfiguration

### Foundation 23.1

Written by: Documentation Team, R&D
Date: Monday, May 22, 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.

# Inhalt


Überblick ... 10
Die Umgebungsvariable Composerdir ... 10
Berechtigungen im Studio und Client ... 10
Konfigurationsdateien ... 19
Konfiguration des Content Composer Studios ... 19
*Das Element appSettings in der Composer.Studio.exe.config* ... *19*
Laden eines Datenbank-Alias aus einer Datei ... 21
Festlegen der Normal.dotm ... 22
Anzeigen von Dokumenten im internen Dokument-Viewer ... 23
Definieren der Vorlage für die Berichtsgenerierung ... 23
Festlegen des Microsoft Word-Ansichtsmodus ... 24
Festlegen des manuellen Variablentyps ... 24
*Das mws-Element in der Composer.Studio.exe.config* ... *25*
*Studio Startparameter* ... *25*
Unterstützte Parameter ... 25
*Informationen zu Startparametern und Benutzerattributen* ... *26*
Definieren eines Startsystem mit dem Attribut StudioStartSystemOId ... 27
Definieren eines Startordners mit dem Benutzerattribut StudioStartFolder ... 28
Die Konfigurationsdatei odin.config ... 28
*Festlegen des TIFF zu XPS Konverters* ... *30*
Konfiguration des Content Composer Clients ... 31
*appSettings in der Composer.Client.exe.config* ... *31*
Konfigurieren der Odin-Ansichten für den Client ... 34
Anzeigen von Dokumenten im internen Dokument-Viewer ... 34
Ändern des Druckzeitpunkts ermöglichen ... 35
Definieren des Block Expand Levels ... 35
Anzeigen der Prozessbeschreibung ... 35
Laden des Inhalts des Root-Ordners ... 36
Festlegen des Ausgabeorts der MWS-Server-Fehlermeldungen ... 36
Anzeigen einer Meldung bei automatischer Weiterleitung eines Prozesses ... 37
Aktualisieren des Inhaltsverzeichnisses ... 37

Festlegen des Microsoft Word-Ansichtsmodus ... 38
Anzeigen des Dialogs Prozessoptionen ... 38
Einfügen eines Wasserzeichens ... 39
*Informationen zu den Client Startparametern und Benutzerattributen* ... *39*
Definieren eines Startsystem mit dem Attribut StartSystemOId ... 40
Definieren eines Startordners mit dem Benutzerattribut StartFolder ... 40
Konfiguration der Windows-Dienste ... 41
*Konfigurieren der Wartezeit eines Windows-Dienstes* ... *41*
*Netzlaufwerke* ... *42*
Festlegen des Pfades zu einem Netzlaufwerk ... 42
Informationen zu Laufwerkszuordnungen ... 42
Zuordnen eines Laufwerksbuchstabens zu einem Netzlaufwerk ... 42
*Konfiguration der MWS-Dienste* ... *43*
Das Element appSettings in der Composer.MWS.exe.config ... 43
Keys ... 43
Festlegen der Normal.dotm ... 45
Festlegen des manuellen Variablentyps ... 45
Das mws Element in der Composer.MWS.exe.config ... 46
Attribute des Elements mws ... 46
Attribute des Elements serviceuser ... 47
Konfigurieren des Dokumentenschutzes ... 48
Definieren von Service-Usern ... 49
Verwenden einer Pool-Variablen als Prozesstitel ... 50
Handhaben fehlender Content Controls ... 50
Ausführen des Composer.MWS-Dienstes ohne Administratorrechte ... 51
Konfiguration der Kommunikation ... 53
*Informationen zur Content Composer Kommunikation* ... *53*
*HTTPS-Konfiguration für den WCF-Kommunikationskanal* ... *54*
Installieren des SSL-Zertifikats ... 54
Aktivieren von HTTPS für den WCF-Kommunikationskanal ... 55
Importieren der RSA Encryption Keys auf dem Client ... 56

Vergeben von Zugriffsrechten für den RSA-Container ..... 56
Problembehebung ..... 57
*Security Token Service (sts)* ..... *57*
Konfigurieren des Security Token Services (sts) auf dem Server ..... 57
Konfigurieren des Security Token Services (sts) auf dem Client ..... 57
*Composer User Repository (mur)* ..... *58*
Konfigurieren des Composer User Repositorys (mur) auf dem Server ..... 58
Konfigurieren des Composer User Repositorys (mur) auf dem Client ..... 58
*DataProviderService (dpws)* ..... *59*
Konfigurieren des DataProviderServices (dpws) auf dem Server ..... 59
Konfigurieren des DataProviderServices (dpws) auf dem Client ..... 59
*RepositoryWebService (rws)* ..... *60*
Konfigurieren des RepositoryWebServices (rws) auf dem Server ..... 60
Konfigurieren des RepositoryWebServices (rws) auf dem Client ..... 60
*OdinWebService (ows und owsbasic)* ..... *60*
Konfigurieren des OdinWebServices (ows) auf dem Server ..... 60
Konfigurieren des OdinWebServices (ows) auf dem Client ..... 61
Client-Konfiguration owsbasic ..... 61
Konfigurieren des OdinWebServices (owsbasic) auf dem Server ..... 62
*XdataWebService (xws und xwsbasic)* ..... *62*
Konfigurieren des XdataWebServices (xws) auf dem Server ..... 62
Konfigurieren des XdataWebServices (xws) auf dem Client ..... 62
Client-Konfiguration xwsbasic ..... 63
Konfigurieren des XdataWebServices (xwsbasic) auf dem Server ..... 63
*ComposerWebService (mws und mwsbasic)* ..... *64*
Konfigurieren des ComposerWebServices (mws) auf dem Server ..... 64
Konfigurieren des ComposerWebServices (mws) auf dem Client ..... 64
Client-Konfiguration mwsbasic ..... 64
Konfigurieren des ComposerWebServices (mwsbasic) auf dem Server ..... 65
*ComposerRepositoryWebService (mwsrws und mwsrwsbasic)* ..... *65*
Konfigurieren des ComposerRepositoryWebServices (mwsrws) auf dem Server ..... 65
Konfigurieren des ComposerRepositoryWebServices (mwsrws) auf dem Client ..... 66
Client-Konfiguration mwsrwsbasic ..... 66

Konfigurieren des ComposerRepositoryWebServices (mwsrwsbasic) auf dem Server ..... 66
*RemoteControlService* ..... *67*
Informationen zu Remote Control ..... 67
Client Konfiguration (Studio) ..... 67
Server-Konfiguration ..... 69
*RemoteDebugger* ..... *69*
Informationen zum Debug Service RemoteDebugger ..... 69
Client-Konfiguration (Client) ..... 70
Server-Konfiguration ..... 71
Content Composer Zertifikate ..... 71
*Informationen zu Zertifikaten* ..... *71*
*Austauschen der Zertifikate* ..... *72*
Entfernen des Root-Zertifikats composer.CA ..... 72
Erstellen der Zertifikate ..... 72
Installieren der Zertifikate ..... 73
Anpassen der Server-Konfiguration ..... 74
Anpassen der Studio-Konfiguration ..... 75
Anpassen der Client-Konfiguration ..... 76
*Prüfen der Funktionalität* ..... *76*
Konfiguration des Loggings ..... 77
*Informationen zum Logging* ..... *77*
*Informationen zum .NET Framework Logging* ..... *77*
Aktivieren des .NET Framework-Loggings ..... 77
Standard-Speicherort der Logdatei ..... 77
Ändern des Speicherorts der Log-Dateien ..... 77
Ändern des Log-Levels ..... 78
*Informationen zum Hyland Logging* ..... *79*
Aktivieren des Hyland-Loggings ..... 79
Routen-Konfiguration ..... 79
Profile ..... 81
Verfügbare Route-Konfigurationen ..... 82
*Informationen zum Audit Logging* ..... *82*

Aktivieren des Audit Loggings ....82
Audit Logging Konfiguration ....83
Protokollierte Benutzeraktionen im Studio ....85
Protokollierte Benutzer-Aktionen im Web Client ....87
Protokollierte Odin-Benutzer-Aktionen ....87
Deaktivieren des Audit Loggings ....89
*Informationen zum Splunk Logging* ....*90*
Konfigurieren von Content Composer für Splunk ....90
Splunk Route Konfigurations-Keys ....91
Konfiguration der Benutzerverwaltung ....91
*Die Benutzerverwaltung* ....*91*
*Konfigurieren der Windows-Authentifizierung mit einer Workgroup oder einem Domain-Controller* ....*92*
*Konfigurieren der Windows-Authentifizierung durch einen Domaincontroller mit LDAP* ....*92*
*Aktivieren der Windows-Authentifizierung* ....*93*
*Konfigurieren der LDAP-Authentifizierung* ....*94*
*Anpassen der allgemeinen User Repository-Einstellungen* ....*95*
Beispiel-Konfiguration ....97
*Anpassen der User-Repository-Einstellungen für LDAP* ....*97*
Beispiel-Konfiguration ....100
Verfügbare Standardprozess ....100
*Der Prozess MWS Cleanup* ....*100*
Informationen zum MWS Cleanup-Prozess ....100
Importieren des MWS Cleanup-Prozesses ....101
*Der Prozess Odin Cleanup* ....*102*
Informationen zum Odin Cleanup-Prozess ....102
Importieren des Odin Cleanup-Prozesses ....102
*Der Prozess OdinOnBaseArchive* ....*102*
Informationen zum OdinOnBaseArchive-Prozess ....102
Importieren der OdinOnBaseArchive-Prozesses ....103
*Der Prozess ScheduledTransfers* ....*103*
Informationen zum ScheduledTransfers-Prozess ....103
Importieren des ScheduledTransfer-Prozesses ....104
Prozesse hosten ....104

*Installation eines Windows-Services* ... *104*
Kopieren der erforderlichen Dateien ... 104
Anpassen der neu erstellten Konfigurationsdatei ... 105
Festlegen der Prozesse ... 105
Registrieren der Dienste ... 107
*Anpassen der Konfigurationsdateien* ... *108*
*Deinstallation eines Windows-Dienstes* ... *110*
Datenbanken für Volltextsuche einrichten ... 111
*Oracle* ... *111*
*SQL Server* ... *111*
Importieren eines Transferjobs ... 112
Informationen zur System-Zugriffssteuerung ... 113
*Aktivieren der System-Zugriffssteuerung* ... *113*
Informationen zum Cache Management ... 114
*Aktivieren des Cache Managements* ... *114*
Informationen zur OnBase Document REST API ... 116
*Verbindung zur OnBase Document REST API* ... *116*
*Anpassen der IdP-Konfigurationsdatei* ... *119*
Load Balancer Konfiguration ... 119
*Voraussetzungen* ... *120*
*Sichern der Konfigurationsdateien* ... *121*
*Umgebungsvariablen* ... *121*
*Installieren des Zertifikats* ... *121*
*IP Network Port Konfiguration* ... *122*
Anpassen der Composer.Core.exe.config zur Verwendung eines IP-Ports ... 122
Anpassen der Composer.MWS.exe.config zur Verwendung eines IP-Ports ... 124
Anpasen der Composer.OWS.exe.config zur Verwendung eines IP-Ports ... 126
Anpassen der Composer.XWS.exe.config zur Verwendung eines IP-Ports ... 128
Ändern der Composer.WindowsServiceHost.exe.config für die Verwendung eines IP-Ports ... 129
Anpassen der Composer.Studio.exe.config zur Verwendung eines IP-Ports ... 129
*HTTPS Konfiguration für Load Balancer* ... *130*
Anpassen der Composer.Core.exe.config zur Verwendung von HTTPS ... 130
Anpassen der Composer.MWS.exe.config zur Verwendung von HTTPS ... 134

Anpassen der Composer.OWS.exe.config zur Verwendung von HTTPS ..... 136
Anpassen der Composer.XWS.exe.config zur Verwendung von HTTPS ..... 137
Anpassen der Composer.WindowsServiceHost.exe.config zur Verwendung von HTTPS ..... 138
Anpassen der Composer.Studio.exe.config zur Verwendung von HTTPS ..... 139
*Geteilte Konfigurationsdateien ..... 140*
Einrichten der ersten Umgebung ..... 140
Einrichten einer zusätzlichen Umgebung ..... 142
*Verwenden des Studio über den Load Balancer Server ..... 144*
*Content Composer Web Client-Konfiguration für den Load Balancer ..... 148*
Vorbereiten des Content Composer Rest API Servers ..... 148
Installation des Content Composer Web Clients ..... 149
*Beispiel für eine Logging-Konfiguration ..... 149*
Beispiel einer Frontend-Load-Balancer-Konfiguration für Content Composer Web Client ..... 149
Beispiel einer Backend-Load-Balancer-Konfiguration für Content Composer Web Client ..... 149
Beispiel für eine Frontend-Load-Balancer-Konfiguration für Content Composer Studio ..... 150
Beispiel einer Backend-Load-Balancer-Konfiguration für Content Composer Studio ..... 150

# Überblick

Dieses Handbuch enthält Informationen darüber, wie Sie Ihr Content Composer-System nach der Installation anpassen können.

Die Informationen in diesem Handbuch richten sich an erfahrene Benutzer, die mit Content Composer vertraut sind.

# Die Umgebungsvariable *Composerdir*

Das Content Composer Setup erstellt die Umgebungsvariable **Composerdir**, die den Pfad zum Content Composer Installationsverzeichnis enthält.

**Composerdir** ist eine Systemvariable und daher für alle Microsoft Windows-Benutzer sichtbar.

**Beispiel**

Wurde Content Composer im Verzeichnis *D:\ContentComposer* installiert, hat die Umgebungsvariable den Wert *D:\ContentComposer*.

**Hinweis**: Der Wert der Umgebungsvariable endet mit einem Backslash.

# Berechtigungen im Studio und Client

Dieses Kapitel listet alle Rechte auf, die Content Composer Benutzern zugewiesen werden können.

**Die folgende Tabelle listet die allgemeinen Rechte auf:**

| Allgemeine Rechte | Nummer | Beschreibung | Studio Berechtigung | Client Berechtigung |
|---|---|---|---|---|
| Studio ausführen | 1 | Studio ausführen | • | |
| Navigator anzeigen | 2 | Navigator anzeigen | • | |
| Client Navigator anzeigen | 3 | Navigator im Client anzeigen<br>Der Navigator wird nicht angezeigt, wenn der Client extern gestartet wird. | | • |
| Allgemeine Einstellungen | 4 | Der Benutzer kann das Paging im Navigator aktivieren / deaktivieren.<br>(Hauptmenü->Optionen -> Allgemeine Einstellungen) | • | |

| Allgemeine Rechte | Nummer | Beschreibung | Studio Berechtigung | Client Berechtigung |
|---|---|---|---|---|
| System wechseln | 5 | Ermöglicht den Systemwechsel. | • | |
| Client Systemwechsel | 6 | Ermöglicht den Systemwechsel im Client.<br>**Hinweis**: Die Berechtigung *Client Systemwechsel* setzt die Berechtigung #3 *Client Navigator anzeigen* voraus. | | • |
| Ordner wechseln | 7 | Die Berechtigung wird nur geprüft, wenn ein Startordner angegeben wurde.<br>Ermöglicht den Wechsel in untergeordnete Verzeichnisse, jedoch nicht in übergeordnete Verzeichnisse.<br>Wurde kein Startordner angegeben, ist der Navigator leer und es wird eine entsprechende Meldung angezeigt. | • | |
| Filter- und Suchfunktion | 8 | Filtern von Objektnamen und Volltextsuche wird im aktuellen System ermöglicht.<br>Fehlt das Recht, werden im Navigator die Reiter **Filtern** und **Suchen** ausgeblendet. | • | |
| Objekttypfilter anzeigen | 9 | Anzeige des Auswahlfilters für die Anzeige der verschiedenen Objekttypen im Navigator | • | |

| Allgemeine Rechte | Nummer | Beschreibung | Studio Berechtigung | Client Berechtigung |
|---|---|---|---|---|
| Systeme übertragen | 10 | Systeme können transferiert werden | • | |
| Systeme synchronisieren | 11 | Systeme können synchronisiert werden | • | |
| Administration | 12 | Durchführen von Administrationsaufgaben. Fehlt das Recht, ist der Reiter Administration im Ribbon nicht sichtbar.<br>Administrationsmodule können nicht gestartet werden. | • | |
| Erzeugte Dokumente herunterladen | 13 | Erzeugte Dokumente herunterladen | • | • |
| Prozessliste | 14 | Prozessliste anzeigen | | • |
| Weitergeleitete Prozessliste | 15 | Liste der weitergeleiteten Prozesse anzeigen. | | • |
| Dokument manuell einfügen | 16 | Manuelles Einfügen von zusätzlichen Dokumenten erlauben | | • |
| Baustein manuell einfügen | 17 | Manuelles Einfügen von zusätzlichen Bausteinen erlauben | | • |
| Drucker für alle Dokumente im Bündel setzen | 18 | Zuweisen oder Ändern des Druckers für alle Dokumente im Bündel | | • |
| Drucker für Dokument setzen | 19 | Zuweisen oder Ändern des Druckers für einzelne Dokumente | | • |
| Druckparameter Assistent | 20 | Erlaubt die Einstellung von Ausgabeparametern | | • |

| Allgemeine Rechte | Nummer | Beschreibung | Studio Berechtigung | Client Berechtigung |
|---|---|---|---|---|
| Prozess manuell weiterleiten | 21 | Prozesse manuell an andere Benutzer oder Benutzergruppen weiterleiten | | • |
| Dokumente erstellen | 22 | Dokumente erstellen | | • |
| Datenbeschaffung | 23 | Datenbeschaffung ausführen | | • |
| Drucken und Archivieren | 24 | Drucken und Archivieren erzeugter Bündel | | • |
| Client Administration | 25 | Durchführen von Administrationsaufgaben. Fehlt das Recht, ist der Reiter Administration im Ribbon nicht sichtbar.<br>Administrationsmodule können nicht gestartet werden. | | • |
| Prozessliste für angegebenen Benutzer | 26 | Ermöglicht das Anzeigen der Prozessliste des im Parameter *listuser* hinterlegten Benutzers | | • |
| Client Ordnerwechsel | 27 | Die Berechtigung wird nur geprüft, wenn ein Startordner angegeben wurde. Ermöglicht den Wechsel in untergeordnete Verzeichnisse, jedoch nicht in übergeordnete Verzeichnisse.<br>Wurde kein Startordner angegeben, ist der Navigator leer und es wird eine entsprechende Meldung angezeigt. | | • |
| Client Ansicht Odin | 28 | Odin-Prozesse anzeigen | | • |

| Allgemeine Rechte | Nummer | Beschreibung | Studio Berechtigung | Client Berechtigung |
|---|---|---|---|---|
| Prozesse | | | | |
| Client Ansicht Odin Stapel | 29 | Odin-Stapel anzeigen | | • |
| Client Ansicht Odin offene Kuverts | 30 | Offene Kuverts anzeigen | | • |
| Client Ansicht offene Jobs | 31 | Offene Jobs anzeigen | | • |
| Cache des MWS-Server leeren | 32 | Cache des MWS-Server leeren | | • |
| externe Dokumentdatei zuordnen/lösen | 33 | Ersetzen von Dokumenten im Bündel durch extern gespeicherte Dokumentdateien. | | • |
| externe Bausteindatei zuordnen/lösen | 34 | Ersetzen von Bausteinen in Dokumenten durch extern gespeicherte Dokumentdateien. | | • |
| Erzeugtes Dokument speichern unter… | 35 | Speichern erzeugter Dokumente in einem Verzeichnis. Diese Funktionalität erfordert anwenderseitig auch das Recht<br>**Erzeugtes Dokument herunterladen (13).** | | • |
| Release-Version erstellen | 36 | Erstellen neuer Release-Versionen | • | |
| Objekt endgültig löschen | 37 | Endgültiges Löschen von Objekten aus dem Papierkorb | • | |
| Berichte | 38 | Erstellen von objektübergreifenden | • | |

| Allgemeine Rechte | Nummer | Beschreibung | Studio Berechtigung | Client Berechtigung |
|---|---|---|---|---|
| | | Berichten.<br>Fehlt das Recht, ist der Reiter Berichte im Ribbon nicht sichtbar. Auf Systemberichte kann nicht zugegriffen werden. | | |
| Client Dokumentvorlagen editieren | 39 | Dokumentvorlagen im Client editieren | | • |
| Client Textbausteine editieren | 40 | Textbausteine im Client editieren | | • |
| Gesperrte Objekte anzeigen | 41 | Gesperrte Objekte anzeigen | • | |
| Work-Versionen anzeigen | 42 | Work-Versionen anzeigen | • | |
| Geplante Transfers anzeigen | 43 | Geplante Transfers anzeigen | • | |
| Transferhistorie anzeigen | 44 | Transferhistorie anzeigen | • | |
| Gelöschte Objekte anzeigen | 45 | Gelöschte Objekte anzeigen | • | |
| Web Client Kommunikation erstellen | 46 | Ermöglicht das Erstellen einer Kommunikation im Content Composer Web Client | | • |
| Historische Objektversion öffnen | 47 | Ermöglicht das Öffnen und Reaktivieren einer alten Objektversion | • | |
| Odin Ansichten im Studio | 48 | Ermöglicht den Zugriff auf die Odin-Ansichten im Content Composer Studio | • | |

| Allgemeine Rechte | Nummer | Beschreibung | Studio Berechtigung | Client Berechtigung |
|---|---|---|---|---|
| Abfrage Assistent | 49 | Ermöglicht den Zugriff auf den Abfrage-Assistenten | • | |
| DB-Alias Verwaltung | 50 | Ermöglicht den Zugriff auf die DB Alias Verwaltung | • | |
| Systemverwaltung | 51 | Ermöglicht den Zugriff auf die Systemverwaltung | • | |
| RemoteControl | 52 | Ermöglicht den Zugriff auf Remote Control | • | |
| Odin Import Procedure Manager | 53 | Ermöglicht den Zugriff auf den Odin Import Procedure Manager | • | |
| Web Client Kommunikation löschen | 54 | Ermöglicht das Löschen einer Kommunikation im Content Composer Web Client | | • |
| Testen von Bündeln und Dokumenten im Studio | 3509 | Benutzer darf im Studio Dokumente und Bündel im integrierten Client testen | • | |

**Die folgende Tabelle listet die Berechtigungen für die Odin-Prozesse auf:**

| Odin-Prozess Berechtigungen | Nummer | Beschreibung | Studio Berechtigung | Client Berechtigung |
|---|---|---|---|---|
| Odin-Prozess entsperren | 3000 | Der Benutzer kann einen gesperrten Prozess entsperren (Unlock) | • | • |
| Odin Prozess zurücksetzen | 3001 | Benutzer darf Prozesse von **Gesperrt** auf **Rendition erzeugt** oder **Importiert** | • | • |

| Odin-Prozess Berechtigungen | Nummer | Beschreibung | Studio Berechtigung | Client Berechtigung |
|---|---|---|---|---|
| | | zurücksetzen. | | |
| Odin-Stapel zum Löschen/Drucken in der Odin-Ansicht **Prozess** erstellen | 3002 | Der Benutzer kann aus den Prozess-Jobs einen Stapel zum Drucken oder Löschen erstellen | • | • |
| Odin Prozess löschen | 3003 | Benutzer darf Prozesse auf Status **Löschen** setzen. | • | • |
| Odin Prozess pausieren | 3004 | Der Benutzer kann einen Prozess sperren | • | • |
| Odin-Stapel zum Löschen oder Drucken in der Odin-Ansicht **Offene Jobs** erstellen | 3005 | Der Benutzer kann aus den Offenen Jobs einen Stapel zum Drucken oder Löschen erstellen | • | • |

**Die folgende Tabelle listet die Berechtigungen für die Odin-Ansicht Stapel auf:**

| Berechtigungen Odin Ansicht Stapel | Nummer | Beschreibung | Studio Berechtigung | Client Berechtigung |
|---|---|---|---|---|
| Odin Stapel editieren | 3500 | Der Benutzer kann Stapel bearbeiten | • | • |
| Odin Stapel freigeben | 3501 | Der Benutzer kann Stapel freigeben | • | • |
| Odin Stapel löschen | 3502 | Der Benutzer kann Stapel löschen | • | • |
| Odin Stapel zurücksetzen | 3503 | Der Benutzer kann Stapel auf den Status **Warten** zurücksetzen | • | • |

| Berechtigungen Odin Ansicht Stapel | Nummer | Beschreibung | Studio Berechtigung | Client Berechtigung |
|---|---|---|---|---|
| Odin Stapel auf 'Streaming' zurücksetzen | 3504 | Der Benutzer kann Stapel auf den Status **Streaming** zurücksetzen | • | • |
| Odin Stapel sperren | 3505 | Der Benutzer kann Stapel sperren | • | • |
| Odin Stapel entsperren | 3506 | Der Benutzer kann Stapel entsperren | • | • |
| Odin Stapel zurücksetzen auf | 3507 | Der Benutzer kann Stapel auf einen bestimmten Stapeltyp zurücksetzen | • | • |
| Odin Job Status setzen | 3508 | Der Benutzer kann Jobs auf einen bestimmten Status zurücksetzen | • | • |
| Odin Job Drucker setzen | 3509 | Der Benutzer kann den Drucker des Jobs ändern | • | • |
| Odin Kuvert aus Stapel entfernen | 3510 | Der Benutzer kann ein Kuvert aus dem Stapel entfernen | • | • |
| Odin Job aus Stapel/Kuvert entfernen | 3512 | Der Benutzer kann einen Job aus einem Kuvert/Stapel entfernen | • | |

# Konfigurationsdateien

Die Content Composer Konfigurationsdateien befinden sich im gleichen Verzeichnis wie die jeweilige Anwendung.

**Die folgende Tabelle listet die relevanten Konfigurationsdateien der Content-Composer-Anwendungen und -Dienste auf:**

| Programm / Dienst | Konfigurationsdatei |
|---|---|
| Studio | Composer.Studio.exe.config |
| Windows Client | Composer.Client.exe.config |
| Core Dienst | Composer.Core.exe.config |
| MWS Dienst | Composer.MWS.exe.config |
| OWS Dienst | Composer.OWS.exe.config |
| XWS Dienst | Composer.XWS.exe.config |
| Benutzerdefinierter Dienst | UserDefined.exe.config |
| Repository Setup | RepositorySetup.exe.config |
| Content Composer Object Tool | Composer.Tool.exe.config |
| DOCX-Compiler | DocXCompiler.exe.config |
| Runtime Console Host | Composer.ConsoleHost.exe.config |
| Windows Service Host | Composer.WindowsServiceHost.exe.config |
| WCF-Console | Composer.WCF.Console.exe.config |
| Transfer Console | TransferConsole.exe.config |
| Monalisa Engine | MonalisaEngine.exe.config |

# Konfiguration des Content Composer Studios

## Das Element *appSettings* in der Composer.Studio.exe.config

**Die folgende Tabelle listet die Attribute des appSettings-Elements auf:**

| Attribut | Beschreibung |
|---|---|
| loadDbAliasFromLocalFile | Weitere Informationen finden Sie unter Laden eines Datenbank-Alias aus einer Datei |
| wordtemplate | Weitere Informationen finden Sie unter Festlegen der Normal.dotm. |
| sts | Weitere Informationen finden Sie unter Konfigurieren des Security Token Services (sts) auf dem Client . |
| login | Nur zur internen Verwendung. |
| license | Nur zur internen Verwendung. |
| rws | Weitere Informationen finden Sie unter Konfigurieren des RepositoryWebServices (rws) auf dem Client. |
| mwsrws | Weitere Informationen finden Sie unter Konfigurieren des ComposerRepositoryWebServices (mwsrws) auf dem Client. |
| dpws | Weitere Informationen finden Sie unter Konfigurieren des DataProviderServices (dpws) auf dem Client. |
| odinparameterfile | Gibt den Namen der Konfigurationsdatei an, in der die Parameter für den Import von Druckjobs in die Odin-Datenbank definiert sind.<br>Weitere Informationen finden Sie unter "Übergeben von Daten aus einem Dokument in die Tabelle ODIN_JOB", "Übergeben von Daten aus einem Bündel in die Tabelle ODIN_JOB" und "Erweitern der Stored Procedure INSERT_JOB" in der Content Composer Studio Hilfe. |
| ows | Weitere Informationen finden Sie unter Konfigurieren des OdinWebServices (ows) auf dem Client. |
| useInternalDocumentViewer | Weitere Informationen finden Sie unter Anzeigen von Dokumenten im internen Dokument-Viewer. |
| reportwordtemplate | Weitere Informationen finden Sie unter Definieren der Vorlage für die Berichtsgenerierung. |
| WordForcePrintView | Weitere Informationen finden Sie unter Festlegen des Microsoft Word-Ansichtsmodus. |
| manVarResultXmlElementType | Definiert den Typ des Elements für manuelle Variablen, das im XML |

| Attribut | Beschreibung |
|---|---|
| | zurückgegeben wird.<br>**Mögliche Werte**<br>• **manvar**: Nur der Typ 'manvar' ist im Info XML enthalten.<br>• **manvar2**: Nur der Typ 'manvar2' ist im Info XML enthalten.<br>• **both**: Beide Typen sind in Info XML enthalten..<br>Fehlt der Key oder ist der angegebene Wert außerhalb des gültigen Bereichs, wird als Standardwert **both** verwendet. |

**Beispiel**

```
<appSettings>
   <add key="loadDbAliasFromLocalFile" value="FALSE" />
   <add key="wordtemplate" value="%composerdir%normal.dotm"/>
   <add key="sts" value="http://localhost:8000/sts"/>
   <add key="login" value="http://localhost:8010/mur/login"/>
   <add key="license" value="http://localhost:8010/mur/license"/>
   <add key="rws" value="http://localhost:8010/rws"/>
   <add key="mwsrws" value="http://localhost:8010/mws/mwsrepository"/>
   <add key="dpws" value="http://localhost:8010/dataprovider"/>
   <add key="odinparameterfile" value="odinparameter.config"/>
   <add key="ows" value="http://localhost:8010/ows/owsrepository"/>
   <add key="useInternalDocumentViewer" value="FALSE" />
   <add key="reportwordtemplate"
value="D:\ContentComposer\Templates\Report.docx"/>
   <add key="WordForcePrintView" value="TRUE" />
   <add key="manVarResultXmlElementType" value="both" />
</appSettings>
```

## Laden eines Datenbank-Alias aus einer Datei

Um einen Datenbank-Alias aus einer Datei zu laden, führen Sie die folgenden Schritte durch.

**Prerequisite**

Eine XML-Datei mit der erforderlichen Struktur. Informationen zum Erstellen einer solchen Datei finden Sie unter "Exportieren eines Datenbank-Alias" in der *Content Composer Studio-Hilfe*.

**Hinweis**:

Soll der Datenbank-Alias aus einer lokalen Datei geladen werden, muss der Alias auch in dem Repository-System vorhanden sein, auch wenn er nicht von dort geladen werden soll.

Wird ein Datenbank-Alias aus einer Datei geladen, wird eine entsprechende Meldung auf dem Level *Information* protokolliert.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Studio.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die Zeile, die wie folgt beginnt:

   ```
   <add key="loadDbAliasFromLocalFile"
   ```

3. Führen Sie einen der folgenden Teilschritte durch.
   - Um den Datenbank-Alias aus einer lokalen Datei zu laden, setzen Sie den Wert des `value` Attributs auf `TRUE`.
   - Um den Datenbank-Alias aus dem Repository zu laden, setzen Sie den Wert des `value` Attributs auf `FALSE`.

   **Beispiel**

   ```
   <add key="loadDbAliasFromLocalFile" value="TRUE" />
   ```

   **Hinweis**: Der Defaultwert ist `FALSE`.
4. Suchen Sie nach dem `<configSections>` Element.
5. Fügen Sie innerhalb des Elements `<configSections>` die folgende Zeile hinzu.

   ```
   <section name="mlsdatabase"
   type="ComposerSuite.Common.SystemFramework.AliasConfigurationSection,
   ComposerSuite.Common.SystemFramework"/>
   ```

6. Fügen Sie unterhalb des Elements `</configuration>` die folgende Zeile ein und ändern Sie `CustomAlias.xml` in den gewünschten Dateinamen.

   ```
   <mlsdatabase configSource="CustomAlias.xml" />
   ```

7. Speichern und schließen Sie die Datei.

## Festlegen der Normal.dotm

Um festzulegen, welche Normal.dotm Content Composer zum Generieren der Dokumente verwendet, führen Sie die folgenden Schritte aus:

Weitere Informationen finden Sie unter "Word-Dokumentvorlage zuweisen" und "Informationen zu Word-Dokumentvorlagen" in der *Content Composer Studio Hilfe*.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Studio.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die Zeile, die wie folgt beginnt:

   ```
   <add key="wordtemplate"
   ```

3. Setzen Sie das Attribut `value` auf den gewünschten Pfad und Dateinamen.

   **Beispiel**

```
<add key="useInternalDocumentViewer" value="C:\Templates\Normal.dotm"
/>
```

**Hinweis**: Der Defaultwert ist `%Composerdir%Normal.dotm`.

4. Speichern und schließen Sie die Datei.

### Anzeigen von Dokumenten im internen Dokument-Viewer

Um generierte Dokumente im internen Dokument-Viewer statt in Word anzuzeigen, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Studio.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die Zeile, die wie folgt beginnt:

   ```
   <add key="useInternalDocumentViewer"
   ```

3. Führen Sie einen der folgenden Teilschritte durch.
   - Um generierte Dokumente im internen Dokument-Viewer anzuzeigen, setzen Sie das `value` Attribute auf `True`.
   - Um generierte Dokumente in Word anzuzeigen, setzen Sie das `value` Attribut auf `False`.

   **Beispiel**

   ```
   <add key="useInternalDocumentViewer" value="True" />
   ```

   **Hinweis**: Die Default-Einstellung ist `False`.

4. Speichern und schließen Sie die Datei.

### Definieren der Vorlage für die Berichtsgenerierung

Um die DOCX-Datei festzulegen, die Content Composer Studio als Vorlage für die Berichterstellung verwendet, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Studio.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die Zeile, die wie folgt beginnt:

   ```
   <add key="reportwordtemplate"
   ```

3. Setzen Sie das Attribut `value` auf den gewünschten Pfad und Dateinamen.

   **Beispiel**

   ```
   <add key="reportwordtemplate" value="C:\Templates\Report.docx" />
   ```

   **Hinweis**: Der Defaultwert ist `"%Composerdir%Report.docx`.

4. Speichern und schließen Sie die Datei.

## Festlegen des Microsoft Word-Ansichtsmodus

Um den Ansichtsmodus festzulegen, den Microsoft Word verwendet, wenn der Benutzer eine Dokumentvorlage oder einen Textblock öffnet, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Studio.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die Zeile, die wie folgt beginnt:

   ```
   <add key="WordForcePrintView"
   ```

3. Führen Sie einen der folgenden Teilschritte durch.
   - Um generierte Dokumente in Microsoft Word in der Druckansicht anzuzeigen, setzen Sie das `value` Attribute auf `True`.
   - Um generierte Dokumente in Microsoft Word nicht in der Druckansicht anzuzeigen, setzen Sie das `value` Attribute auf `False`.

   **Beispiel**

   ```
   <add key="WordForcePrintView" value="True" />
   ```

   **Hinweis**: Der Defaultwert ist `FALSE`.

4. Speichern und schließen Sie die Datei.

## Festlegen des manuellen Variablentyps

Um den Elementtyp für manuelle Variablen festzulegen, der in der XML-Datei zurückgegeben wird, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Studio.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die Zeile, die wie folgt beginnt:

   ```
   <add key="manVarResultXmlElementType"
   ```

3. Führen Sie einen der folgenden Teilschritte durch.
   - Um nur den Typ `manvar` in die Info-XML aufzunehmen, setzen Sie das Attribut `value` auf `manvar`.
   - Um nur den Typ `manvar2` in die Info-XML aufzunehmen, setzen Sie das Attribut `value` auf `manvar2`.
   - Um beide Typen in die Info-XML aufzunehmen, setzen Sie das Attribut `value` auf `both`.

   **Beispiel**

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

   **Hinweis**: Der Defaultwert ist `both`.

4. Speichern und schließen Sie die Datei.

## Das mws-Element in der Composer.Studio.exe.config

**Die folgende Tabelle listet die Attribute des msw-Elements auf:**

| Attribut | Beschreibung |
|---|---|
| systemoid | Objekt-ID des Systems, in dem der MWS-Prozess sowie der Datenbank-Alias gespeichert sind. |
| dbalias | Der Name des Datenbank-Alias, der für den Zugriff auf die MWS-Datenbank verwendet wird. |
| process | Der Name des MWS-Prozesses, der sich in dem im Attribut systemoid angegebenen System befinden muss. |
| client_runtime_ mode | Nur zur internen Verwendung. |

**Beispiel**

```
<mws systemoid="DEBUG OID" dbalias="mws" process="mws_standard" client_
runtime_mode="true"> </mws>
```

## Studio Startparameter

Verwenden Sie die folgenden Parameter, um das Startverhalten des Studios zu konfigurieren:

Die Startparameter haben den folgenden Aufbau.

```
/parametername:Paramvalue
```

**Hinweis**: Enthält der **Paramwert** Leerzeichen, so muss er in Hochkommata angegeben werden.

### Unterstützte Parameter

| Parameter | Beschreibung |
|---|---|
| startsystem | **System Objekt-ID**<br>System-OID des Systems, das nach dem Start automatisch geöffnet werden soll.<br>Fehlt diese Angabe, startet Das Studio mit dem zuletzt ausgewählten System.<br>**Beispiel** |

| Parameter | Beschreibung |
|---|---|
| | `/startsystem:MySystemOId` |
| navipath | **Ordner-Name**<br>Dieses Element definiert, welcher Ordner im Navigator selektiert wird.<br>Wenn kein Startsystem angegeben wurde, bezieht sich die Angabe auf das zuletzt ausgewählte ystem.<br>Der Name des Systems darf nicht im Pfad enthalten sein.<br>**Beispiel**<br>`/navipath:/FolderName` |
| credentials | **Input**: Benutzername und Passwort müssen eingegeben werden.<br>**Silent**: Benutzername und Passwort müssen einmal eingegeben werden. Aktiviert der Benutzer die Login-Option **Passwort speichern**, werden die eingegebenen Daten bei zukünftigen Logins verwendet.<br>**SilentUI**: Verwenden Sie diesen Parameter, um ein **Silent**-Passwort zurückzusetzen. Der Login-Dialog wird wieder angezeigt.<br>**Windows**: Die Anmeldung erfolgt mit dem Windows-Benutzernamen und Passwort. Diese Einstellung wird auch verwendet, wenn der Parameter credentials nicht angegeben ist.<br>**Beispiel**<br>`/credentials:Silent` |

## Informationen zu Startparametern und Benutzerattributen

Die Benutzerattribute **StudioStartSystemOId** und **StudioStartFolder** bieten die Möglichkeit, ein Startsystem und einen Startordner für einen Benutzer oder eine Gruppe von Benutzern zu definieren.

Die Startparameter haben jedoch Vorrang vor den in den Benutzerattributen angegebenen Parametern.

Beachten Sie die folgenden Informationen, wenn Sie Startparameter und Benutzerattribute angeben:

- Wenn der Startparameter **startsystem** gesetzt ist, öffnet das Content Composer Studio das angegebene System.
- Wenn kein Parameter **startsystem**, aber das Benutzerattribut **StudioStartSystemOId** gesetzt ist, öffnet Content Composer Studio das im Benutzerattribut angegebene System.
- Wenn keine der beiden Optionen definiert ist, öffnet Content Composer Studio das zuletzt verwendete System.
- Wurde der Startparameter **navipath** angegeben, werden folgende Prüfungen vorgenommen:
    - Hat der Benutzer das Recht **Ordner wechseln(7)**, wird in den angegebenen Ordner gewechselt.

      Ist der Ordner nicht vorhanden, wird in den Rootfolder des Systems gewechselt.

      Hat der Benutzer das Recht **Ordner wechseln(7)** nicht, wird der angegebene Ordner für den Benutzer der RootFolder des Systems.

      Ist der angegebene Ordner nicht im System vorhanden, wird eine Warnung angezeigt und der Navigator geleert. Der Benutzer kann, wenn er das Recht **System wechseln(5)** hat, in ein anderes System wechseln. Dies ist jedoch nur möglich, wenn ein Ordner mit dem selben Namen im Zielsystem vorhanden ist.
- Wurde der Startparameter navipath nicht angegeben, werden folgende Prüfungen durchlaufen:
    - Ist das UserAttribut **StudioStartFolder** vorhanden und der Benutzer hat das Recht **Ordnerwechsel(7)**, wird keine weitere Aktion ausgeführt.
    - Hat der Benutzer das Recht **Ordner wechseln(7)** nicht, wird der angegebene Ordner für den Benutzer der RootFolder des Systems.

      Ist der angegebene Ordner nicht im System vorhanden, wird eine Warnung angezeigt und der Navigator geleert.

      Der Benutzer kann, wenn er das Recht **System wechseln(5)** hat, in ein anderes System wechseln.

      Dies ist jedoch nur möglich, wenn ein Ordner mit dem selben Namen im Zielsystem vorhanden ist.

### Definieren eines Startsystem mit dem Attribut *StudioStartSystemOId*

Um ein bestimmtes Startsystem für einen Benutzer oder einer Benutzergruppe zu definieren, führen Sie die folgenden Schritte aus.

1. Öffnen Sie im **Content Composer Studio** das entsprechende Profil.
2. Klicken Sie im Reiter **Identity Editor Tools** auf die Schaltfläche **Attribut hinzufügen** .
3. Geben Sie im Fenster **Neues Profilattribut anlegen** den Attributnamen `StudioStartSystemOId` ein und klicken Sie auf **OK**.
4. Geben Sie im **Objektinspektor** für die Eigenschaft **Value** die benötigte System-OID ein.
5. Optional. Um das zu verwendende Benutzerprofil anzugeben, ändern Sie das Attribut **userProfile** in der Datei **UserRepository.config**.

Weitere Informationen finden Sie unter Anpassen der allgemeinen User Repository-Einstellungen.

6. Starten Sie den Dienst **Composer.Core** neu, damit Ihre Änderungen wirksam werden.

### Definieren eines Startordners mit dem Benutzerattribut *StudioStartFolder*

Um einen bestimmten Startordner für einen Benutzer oder einer Benutzergruppe zu definieren, führen Sie die folgenden Schritte aus.

1. Öffnen Sie im **Content Composer Studio** das entsprechende Profil.
2. Klicken Sie im Reiter **Identity Editor Tools** auf die Schaltfläche **Attribut hinzufügen** .
3. Geben Sie im Fenster **Neues Profilattribut anlegen** den Attributnamen `StudioStartFolder` ein und klicken Sie dann auf **OK**.
4. Geben Sie im **Objektinspektor** für die Eigenschaft **Value** den benötigten Startordner ein.

   **Hinweis**: Im Gegensatz zum Startparameter **navipath** wird hierbei nicht der Pfad zu einem Ordner angegeben, sondern der Ordnername selbst. Daher wird dieses Attribut auch nur zur Spezifikation eines Startordners bei fehlendem **MwsChangeFolder**-Recht beachtet.
5. Starten Sie den Dienst **Composer.Core** neu, damit Ihre Änderungen wirksam werden.

## Die Konfigurationsdatei odin.config

```
<?xml version="1.0" encoding="utf-8" ?>
<odinSettings>
   <add key="tempdir" value="%userprofile%\documents"
   <add key="destinationdir" value=""/>
   <add key="printServerPaths" value=""/>
   <add key="compartTraceDir" value="%Composerdir%compartTrace"/>
   <add key="PDFtoXPSConverter" value="Compart"/>
   <add key="TiffConverter" value="Default"/>
   <add key="TiffConverterOutputPageSize" value="A4"/>
</odinSettings>
```

| Attribut | Beschreibung |
|---|---|
| tempdir | Pfad des temporären Odin-Verzeichnisses.<br>Der Standardwert entspricht dem Pfad des temporären Ordners des Windows-Benutzers, dessen Konto zur Ausführung des Windows-Prozesses verwendet wird.<br>Sie können auch Windows-Umgebungsvariablen im Pfad verwenden.<br>**Beispiel**<br>`<add key="tempdir"` |

| Attribut | Beschreibung |
|---|---|
| | `value="%userprofile%\documents"` |
| destinationdir | Pfad des Odin-Zielverzeichnisses.<br>Diese Einstellung hat keinen Standardwert. |
| printServerPaths | Kommaseparierter String, der die Adressen der Druck-Server enthält.<br>Diese Einstellung hat keinen Standardwert. |
| compartTraceDir | Pfad zum compart Traceverzeichnis.<br>Wenn der Windows-Prozess mit einem Windows-Systemkonto ausgeführt wird, lautet der Standardpfad `%Composerdir%`.<br>Wenn der Windows-Prozess mit einem Windows-Benutzerkonto ausgeführt wird, lautet der Standardpfad `%userprofile%\ContentComposer`.<br>Sie können auch Windows-Umgebungsvariablen im Pfad verwenden.<br>**Beispiel**<br>`<add key="compartTraceDir" value="%Composerdir%compartTrace"/>` |
| PDFtoXPSConverter | Gibt die Bibliothek an, die die Komponente **OdinRendition** für die Konvertierung einer PDF-Datei nach XPS verwendet.<br>Derzeit wird nur der Wert `Compart` unterstützt.<br>**Hinweis**: Diese Option muss lizensiert werden. Wenn Sie keine gültige **PdfToXpsCpsdk**-Lizenz besitzen, wenden Sie sich an den technischen Support von Hyland Software. Die Telefonnummern des technischen Supports finden Sie unter www.hyland.com/pswtscontact. Kopieren Sie die Lizenzdatei in das **%Composerdir%** Verzeichnis und stellen Sie sicher, dass der Dateiname **PdfToXpsCpsdk.lic** lautet. |
| TiffConverter | Gibt den Algorithmus an, den die Komponente **OdinRendition** für die Konvertierung einer PDF-Datei nach XPS verwendet.<br>**Mögliche Werte**<br>• **Defaultwert**: Der Standard-TIFF-Konverter wird verwendet.<br>Dies ist der Defaultwert, wenn das Attribut nicht in der Datei `odin.config` vorhanden ist. |

| Attribut | Beschreibung |
|---|---|
| | • **FitToPage**: Der weiterentwickelte TIFF-Konverter wird verwendet.<br>Dieser Konverter bietet die folgenden Funktionen:<br>  • Verkleinert die Bitmap auf die physikalische Druckgröße (shrink to fit).<br>  • Reduziert hohe Farbtiefen auf 8 Bit pro Pixel.<br>  • Reduziert hohe DPI und Bitmap-Abmessungen auf 300 DPI.<br>  • Ermöglicht die Konfiguration der Seitengröße der XPS-Ausgabedatei.<br>Weitere Informationen finden Sie unter Festlegen des TIFF zu XPS Konverters. |
| TiffConverterOutputPageSize | Gibt das Seitenformat der XPS-Ausgabedatei an.<br>**Mögliche Werte**<br>• **A4**: Seitenformat ISO A4. Dies ist der Defaultwert, wenn das Attribut nicht in der Datei `odin.config` vorhanden ist.<br>• **Letter**: Seitenformat Letter, d.h. 8.5 x 11 Zoll / 215.9 x 279.4 mm.<br>Siehe auch Festlegen des TIFF zu XPS Konverters. |

## Festlegen des TIFF zu XPS Konverters

Um den Algorithmus festzulegen, den die Komponente **OdinRendition** zur Konvertierung einer TIFF-Datei nach XPS verwendet, führen Sie die folgenden Schritte aus:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **odin.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die Zeile, die wie folgt beginnt:

   ```
   <add key="TiffConverter"
   ```

3. Um den Standard-TIFF-Konverter zu verwenden, setzen Sie den Wert des Attributs `value` auf `Default`.

   **Beispiel**

   ```
   <add key="TiffConverter" value="Default"/>
   ```

4. Um den weiterentwickelten TIFF-Konverter zu verwenden, führen Sie die folgenden Teilschritte aus.
   1. Setzen Sie das Attribut `value` auf `FitToPage`.

2. Suchen Sie die Zeile, die wie folgt beginnt:

   ```
   <add key="TiffConverterOutputPageSize"
   ```

3. Setzen Sie das Attribut `value` auf einen der folgenden Werte:
   - `A4`: Seitenformat ISO A4.
   - `Letter`: Seitenformat Letter.

**Beispiel**

```
<add key="TiffConverter" value="FitToPage"/>
<add key="TiffConverterOutputPageSize" value="Letter"/>
```

5. Speichern und schließen Sie die Datei.

# Konfiguration des Content Composer Clients

## appSettings in der Composer.Client.exe.config

```
<appSettings>
   <add key="sts" value="http://localhost:8000/sts" />
   <add key="login" value="http://localhost:8010/mur/login" />
   <add key="mur" value="http://localhost:8010/mur/data" />
   <add key="license" value="http://localhost:8010/mur/license" />
   <add key="mwsrws" value="http://localhost:8010/mws/mwsrepository" />
   <add key="mws" value="http://localhost:8011/mws/mwsprocess" />
   <add key="dpws" value="http://localhost:8010/dataprovider" />
   <add key="ows" value="http://localhost:8010/ows/owsrepository"/>
   <add key="odinviewsystemoid" value="dm" />
   <add key="odinviewdbalias" value="odin_mws" />
   <add key="useInternalDocumentViewer" value="FALSE" />
   <add key="enableOnlinePrinterPrintTimeChanging" value="FALSE" />
   <add key="blockExpandLevel" value="1" />
   <add key="shortProcDescLength" value="0" />
   <add key="loadRootFolderOnBundleAutoStart" value="FALSE" />
   <add key="loadFolderTreeOnBundleAutoStart" value="FALSE" />
   <add key="showMessageBoxOnMwsError" value="TRUE" />
   <add key="showInfoBoxOnAutomaticForwarding" value="TRUE" />
   <add key="WordUpdateTOC" value="TRUE" />
   <add key="WordForcePrintView" value="TRUE" />
   <add key="showProcessOptions" value="TRUE" />
   <add key="applyWaterMark" value="waterMark text" />
</appSettings>
```

**In der folgenden Tabelle sind die verfügbaren Keys aufgeführt:**

| Key | Beschreibung |
|---|---|
| sts | Weitere Informationen finden Sie unter Konfigurieren des Security Token Services (sts) auf dem Client. |
| login | Weitere Informationen finden Sie unter Konfigurieren des Composer User Repositorys (mur) auf dem Client. |
| mur | Weitere Informationen finden Sie unter Konfigurieren des Composer User Repositorys (mur) auf dem Client. |
| mwsrws | Weitere Informationen finden Sie unter Konfigurieren des ComposerRepositoryWebServices (mwsrws) auf dem Client. |
| mws | Weitere Informationen finden Sie unter Konfigurieren des ComposerWebServices (mws) auf dem Client. |
| dpws | Weitere Informationen finden Sie unter Konfigurieren des DataProviderServices (dpws) auf dem Client. |
| ows | Weitere Informationen finden Sie unter Konfigurieren des OdinWebServices (ows) auf dem Client. |
| odinviewsystemoid | Weitere Informationen finden Sie unter Konfigurieren der Odin-Ansichten für den Client. |
| odinviewdbalias | Weitere Informationen finden Sie unter Konfigurieren der Odin-Ansichten für den Client. |
| useInternalDocumentViewer | Weitere Informationen finden Sie unter Anzeigen von Dokumenten im internen Dokument-Viewer |
| license | Hostbasierte Adresse des Diensts.<br>Im folgenden Beispiel kommuniziert der Client mit dem Server-Dienst auf dem Computer mit dem Netzwerknamen Hyland am IP-Port 8010:<br>`<add key="license" value="http://Hyland:8010/mur/license" />` |
| enableOnlinePrinterPrintTimeChanging | Weitere Informationen finden Sie unter Ändern des Druckzeitpunkts ermöglichen |

| Key | Beschreibung |
|---|---|
| blockExpandLevel | Weitere Informationen finden Sie unter Definieren des Block Expand Levels |
| shortProcDescLength | Weitere Informationen finden Sie unter Anzeigen der Prozessbeschreibung |
| loadRootFolderOnBundleAutoStart | Weitere Informationen finden Sie unter Laden des Inhalts des Root-Ordners |
| loadFolderTreeOnBundleAutoStart | Diese Einstellung wird nur berücksichtigt, wenn über die Startparameter objectname oder processid ein Bündel übergeben wurde.<br>**TRUE**: Das System wird geladen.<br>**FALSE**: Das System wird nicht geladen, weder der Inhalt des Stammordners, noch die Ordnerstruktur. Die Einstellung von `loadRootFolderOnBundleAutoStart` wird ignoriert. Ein System muss durch eine gültige SystemOID im startsystem-Startparameter definiert sein. Die Auswahl eines Systems ist nur möglich, wenn der Benutzer den aktuellen Prozess schließt oder speichert. Hinweis Wenn der Startparameter `closeonexitprocess` angegeben ist, wird der Client automatisch geschlossen, wenn der Benutzer einen Prozess schließt oder speichert.<br>Der Defaultwert ist TRUE. |
| showMessageBoxOnMwsError | Weitere Informationen finden Sie unter Festlegen des Ausgabeorts der MWS-Server-Fehlermeldungen |
| showInfoBoxOnAutomaticForwarding | Weitere Informationen finden Sie unter Anzeigen einer Meldung bei automatischer Weiterleitung eines Prozesses |
| WordUpdateTOC | Weitere Informationen finden Sie unter Aktualisieren des Inhaltsverzeichnisses |
| WordForcePrintView | Weitere Informationen finden Sie unter Festlegen des Microsoft Word-Ansichtsmodus |
| showProcessOptions | Weitere Informationen finden Sie unter Anzeigen des Dialogs Prozessoptionen |
| applyWaterMark | Weitere Informationen finden Sie unter Einfügen eines Wasserzeichens |

## Konfigurieren der Odin-Ansichten für den Client

Um es Benutzern zu ermöglichen, die Odin-Ansichten im Content Composer Client zu öffnen, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **clientprofile.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Zeile:

   ```
   <ModuleInfo AssemblyFile="ComposerSuite.Odin.OdinViews.dll" />
   ```

3. Falls die Zeile nicht vorhanden ist, fügen Sie sie im Element `<Modules>` ein. Ist die Zeile auskommentiert, entfernen Sie die Kommentarzeichen.
4. Öffnen Sie aus dem Content Composer Installationsverzeichnis die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
5. Um das System zu definieren, aus dem der Datenbank-Alias sowie die Odin-Ansichten geladen werden, führen Sie die folgenden Teilschritte aus.
   1. Suchen Sie die folgenden Zeile:

      ```
      <add key="odinviewsystemoid" value="dm" />
      ```

   2. Ändern Sie bei Bedarf den Wert des Attributs `value`.
6. Um den Namen des Datenbank-Alias anzugeben, führen Sie die folgenden Schritte durch.
   1. Suchen Sie die folgenden Zeile:

      ```
      <add key="odinviewdbalias" value="odin_mws" />
      ```

   2. Ändern Sie bei Bedarf den Wert des Attributs `value`.

## Anzeigen von Dokumenten im internen Dokument-Viewer

Um generierte Dokumente im internen Dokument-Viewer statt in Word anzuzeigen, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die Zeile, die wie folgt beginnt:

   ```
   <add key="useInternalDocumentViewer"
   ```

3. Führen Sie einen der folgenden Teilschritte durch:
   - Um generierte Dokumente im internen Dokument-Viewer anzuzeigen, setzen Sie das `value` Attribute auf `True`.
   - Um generierte Dokumente in Word anzuzeigen, setzen Sie das `value` Attribut auf `False`.

   **Beispiel**

   ```
   <add key="useInternalDocumentViewer" value="True" />
   ```

   **Hinweis**: Die Default-Einstellung ist `False`.

4. Speichern und schließen Sie die Datei.

### Ändern des Druckzeitpunkts ermöglichen

Um Benutzern zu ermöglichen, den Druckzeit für Online-Drucker zu ändern, führen Sie die folgenden Schritte aus:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Fügen Sie innerhalb des Elements `<appSettings>` die folgende Zeile ein:

   ```
   <add key="enableOnlinePrinterPrintTimeChanging" value="True" />
   ```

3. Führen Sie einen der folgenden Teilschritte durch:
   - Um das Ändern des Druckzeitpunkts für Online-Drucker zu ermöglichen, setzen Sie das `value` Attribut auf `True`.
   - Um zu verhindern, dass Benutzer die Druckzeit für Online-Drucker ändern können, setzen Sie das `value` Attribut auf `False`.

   **Hinweis**: Die Default-Einstellung ist `False`.
4. Speichern und schließen Sie die Datei.

### Definieren des Block Expand Levels

Um festzulegen, bis zu welcher Ebene Textbausteine automatisch aufgeklappt werden wenn der Benutzer ein Dokument auswählt, führen Sie die folgenden Schritte aus:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Fügen Sie innerhalb des Elements `<appSettings>` die folgende Zeile ein:

   ```
   <add key="blockExpandLevel" value="1" />
   ```

3. Führen Sie einen der folgenden Teilschritte durch:
   - Um alle Ebenen aufzuklappen, setzen Sie das `value` Attributs auf `-1`.
   - Um alle Textbausteine bis zu einer bestimmten Ebene aufzuklappen, setzen Sie das Attribut `value` auf die gewünschte Ebene, z. B. `3`.
   - Um keine Ebene aufzuklappen, setzen Sie das `value` Attributs auf `0`.

   **Hinweis**: Der Defaultwert ist `1`.
4. Speichern und schließen Sie die Datei.

### Anzeigen der Prozessbeschreibung

Um zu konfigurieren, ob die Prozessbeschreibung in der Prozessliste angezeigt wird, führen Sie die folgenden Schritte aus:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Fügen Sie innerhalb des Elements `<appSettings>` die folgende Zeile ein:

```
<add key="shortProcDescLength" value="0" />
```

3. Führen Sie einen der folgenden Teilschritte durch.
   - Um die Prozessbeschreibung nicht anzuzeigen, setzen Sie das `value` Attribut auf `0`.
   - Um die Prozessbeschreibung anzuzeigen, setzen Sie das `value` Attribut auf einen Wert größer als 0. Die angegebene Zahl definiert die Anzahl der Zeichen, die angezeigt werden.

   **Beispiel**

   ```
   <add key="shortProcDescLength" value="10" />
   ```

   **Hinweis**: Der Defaultwert ist `0`.
4. Speichern und schließen Sie die Datei.

## Laden des Inhalts des Root-Ordners

Um festzulegen, ob der Inhalt des Root-Ordners geladen wird, wenn ein Bündel direkt beim Start des Clients ausgeführt wird, führen Sie die folgenden Schritte aus.

**Hinweis**: Diese Einstellung wird nur berücksichtigt, wenn über die Startparameter `objectname` oder `processid` ein Bündel geladen wird.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Fügen Sie innerhalb des Elements `<appSettings>` die folgende Zeile ein:

   ```
   <add key="loadRootFolderOnBundleAutoStart" value="True" />
   ```
3. Führen Sie einen der folgenden Teilschritte durch.
   - Um den Inhalt des Root-Ordners und die Ordnerstruktur zu laden, setzen Sie das Attribut `value` auf `True`.
   - Um nur die Ordnerstruktur zu laden, setzen Sie den Wert des Attributs `value` auf `False`.

     **Hinweis**: Der Benutzer kann den Inhalt des Stammordners laden, indem er auf **Aktualisieren** klickt.

   **Hinweis**: Die Default-Einstellung ist `True`.
4. Speichern und schließen Sie die Datei.

## Festlegen des Ausgabeorts der MWS-Server-Fehlermeldungen

Um festzulegen, wo die Fehlermeldungen des MWS-Servers angezeigt werden, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Fügen Sie innerhalb des Elements `<appSettings>` die folgende Zeile ein:

   ```
   <add key="showMessageBoxOnMwsError" value="True" />
   ```
3. Führen Sie einen der folgenden Teilschritte durch.

- Um die Fehlermeldungen des MWS-Servers in einem eigenen Fenster anzuzeigen, setzen Sie das `value` Attribut auf `True`.
- Um die Fehlermeldungen des MWS-Servers nur im Ausgabefenster anzuzeigen, setzen Sie das `value` Attribut auf `False`.

**Hinweis**: Die Default-Einstellung ist `True`.

4. Speichern und schließen Sie die Datei.

### Anzeigen einer Meldung bei automatischer Weiterleitung eines Prozesses

Um bei automatischer Weiterleitung eines Prozesse eine Meldung anzuzeigen, führen Sie die folgenden Schritte aus:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Fügen Sie innerhalb des Elements `<appSettings>` die folgende Zeile ein:

```
<add key="showInfoBoxOnAutomaticForwarding"
```

3. Führen Sie einen der folgenden Teilschritte durch.
   - Um die Meldung anzuzeigen, setzen Sie das `value` Attribut auf `True`.
   - Um die nicht Meldung anzuzeigen, setzen Sie das `value` Attribut auf `False`.

   **Hinweis**: Die Default-Einstellung ist `True`.
4. Speichern und schließen Sie die Datei.

### Aktualisieren des Inhaltsverzeichnisses

Um das Inhaltsverzeichnis zu aktualisieren, wenn der Benutzer ein Microsoft Word-Dokument über den Content Composer Client öffnet, führen Sie die folgenden Schritte aus:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die Zeile, die wie folgt beginnt:

```
<add key="WordUpdateTOC"
```

3. Führen Sie einen der folgenden Teilschritte durch.
   - Um das inhaltsverzeichnis zu aktualisieren, setzen Sie das `value` Attribut auf `True`.
   - Um das Inhaltsverzeichnis unverändert zu lassen, setzen Sie den Wert des Attributs `value` auf `False`.

   **Beispiel**

   ```
   <add key="WordUpdateTOC" value="True" />
   ```

   **Hinweis**: Die Default-Einstellung ist `False`.
4. Speichern und schließen Sie die Datei.

## Festlegen des Microsoft Word-Ansichtsmodus

Um den Ansichtsmodus festzulegen, den Microsoft Word verwendet, wenn der Benutzer eine Dokumentvorlage oder einen Textblock öffnet, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Fügen Sie innerhalb des Elements `<appSettings>` die folgende Zeile ein:

   ```
   <add key="WordForcePrintView" value="True" />
   ```

3. Führen Sie einen der folgenden Teilschritte durch.
   - Um generierte Dokumente in Microsoft Word in der Druckansicht anzuzeigen, setzen Sie das `value` Attribute auf `True`.
   - Um generierte Dokumente in Microsoft Word nicht in der Druckansicht anzuzeigen, setzen Sie das `value` Attribute auf `False`.

   **Hinweis**: Der Defaultwert ist `FALSE`.
4. Speichern und schließen Sie die Datei.

## Anzeigen des Dialogs *Prozessoptionen*

Um den Dialog **Prozessoptionen** beim Öffnen eines Bündels anzuzeigen, führen Sie die folgenden Schritte aus:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die Zeile, die wie folgt beginnt:

   ```
   <add key="showProcessOptions"
   ```

3. Führen Sie einen der folgenden Teilschritte durch.
   - Um den Dialog **Prozessoptionen** anzuzeigen, setzen Sie das `value` Attribut auf `True`.
   - Um den Dialog **Prozessoptionen** nicht anzuzeigen, setzen Sie das `value` Attribut auf `False`.

     Für die Werte Prozesstitel und Prozesstyp wird ein Leerstring übergeben.

   **Beispiel**

   ```
   <add key="showProcessOptions" value="True" />
   ```

   **Hinweis**: Die Default-Einstellung ist `True`.
4. Speichern und schließen Sie die Datei.

### Einfügen eines Wasserzeichens

Um ein Wasserzeichen einzufügen, wenn der Benutzer ein mit Microsoft Word erstelltes Dokument in der Vorschau anzeigt oder bearbeitet, führen Sie die folgenden Schritte aus.

Das Wasserzeichen wird eingefügt, bevor das Dokument vom Client geöffnet wird, und wird entfernt, wenn Sie das Dokument schließen.

**Hinweis**: Das Wasserzeichen wird nicht angezeigt, wenn ein Dokument in der Vorschau oder in OnBase angezeigt wird.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Fügen Sie innerhalb des Elements `<appSettings>` die folgende Zeile ein:

   ```
   <add key="applyWaterMark" value="Draft" />
   ```

3. Geben Sie den Text des Wasserzeichens im Attribut `value` an.

   **Hinweis**: Wenn die die Einstellung fehlt oder das Attribut `value` einen Leerstring enthält, wird kein Wasserzeichen eingefügt.

4. Speichern und schließen Sie die Datei.

## Informationen zu den Client Startparametern und Benutzerattributen

Die Benutzerattribute **StartSystemOId** und **StartFolder** bieten die Möglichkeit, ein Startsystem und einen Startordner für einen Benutzer oder eine Gruppe von Benutzern zu definieren.

Die Startparameter haben jedoch Vorrang vor den in den Benutzerattributen angegebenen Parametern.

Beachten Sie die folgenden Informationen, wenn Sie Startparameter und Benutzerattribute angeben:

- Wenn in den Startparameter ein Startsystem angeben ist, öffnet der Client das angegebene System.
- Wenn kein Startparameter, aber das Benutzerattribut **StartSystemOId** angegeben ist, öffnet der Client das im Benutzerattribut angegebene System.
- Wenn keine der beiden Optionen definiert ist, öffnet der Client das zuletzt verwendete System.
- Wurde der Startparameter navipath angegeben, werden folgende Prüfungen vorgenommen:
  1. Wenn der Benutzer über das Recht **Client Ordnerwechsel(27)** verfügt, wird der angegebene Ordner im **Navigator** ausgewählt.

     Wenn der Ordner nicht vorhanden ist, wird im **Navigator** der Rootfolder ausgewählt.

  2. Hat der Benutzer das Recht **Client Ordnerwechsel(27)** nicht, wird der Rootordner des Systems zum Standardordner des Benutzers.

     Ist der angegebene Ordner nicht im System vorhanden, wird eine Warnung angezeigt und der Navigator geleert.

     Der Benutzer kann in ein anderes System wechseln, wenn er das Recht **System wechseln(6)** hat.

Dies ist jedoch nur möglich, wenn ein Ordner mit dem selben Namen im Zielsystem vorhanden ist.

- Wurde der Startparameter navipath nicht angegeben, werden folgende Prüfungen durchlaufen:
  1. Ist das UserAttribut **StartFolder** vorhanden und der Benutzer hat das Recht **Client Ordnerwechsel(27),** wird keine weitere Aktion ausgeführt.
  2. Hat der Benutzer das Recht **Client Ordnerwechsel(27)** nicht, wird der Rootordner des Systems zum Standardordner des Benutzers.

     Ist der angegebene Ordner nicht im System vorhanden, wird eine Warnung angezeigt und der Navigator geleert.

     Der Benutzer kann in ein anderes System wechseln, wenn er das Recht **System wechseln(6)** hat.

     Dies ist jedoch nur möglich, wenn ein Ordner mit dem selben Namen im Zielsystem vorhanden ist.

### Definieren eines Startsystem mit dem Attribut *StartSystemOld*

Um ein bestimmtes Startsystem für einen Benutzer oder einer Benutzergruppe zu definieren, führen Sie die folgenden Schritte aus:

1. Öffnen Sie im **Content Composer Studio** das entsprechende Profil.
2. Klicken Sie im Reiter **Identity Editor Tools** auf die Schaltfläche **Attribut hinzufügen** .
3. Geben Sie im Fenster **Neues Profilattribut anlegen** den Attributnamen `StartSystemOId` ein und klicken Sie auf **OK**.
4. Geben Sie im **Objektinspektor** in der Eigenschaft **Wert** die gewünschte Systemobjekt-ID ein.
5. Starten Sie den Dienst **Composer.Core** neu, damit Ihre Änderungen wirksam werden.

### Definieren eines Startordners mit dem Benutzerattribut StartFolder

Um einen bestimmten Startordner für einen Benutzer oder einer Benutzergruppe zu definieren, führen Sie die folgenden Schritte aus:

1. Öffnen Sie im Content Composer Studio das entsprechende Profil.
2. Klicken Sie im Reiter **Identity Editor Tools** auf die Schaltfläche **Attribut hinzufügen** .
3. Geben Sie im Fenster **Neues Profilattribut anlegen** den Attributnamen `StartFolder` ein und klicken Sie auf **OK**.
4. Geben Sie im **Objektinspektor** in der Eigenschaft **Wert** die gewünschte Systemobjekt-ID ein.

   **Hinweis**: Im Gegensatz zum Startparameter **navipath** wird hierbei nicht der Pfad zu einem Ordner angegeben, sondern der Ordnername selbst. Daher wird dieses Attribut auch nur zur Spezifikation eines Startordners bei fehlendem **MwsChangeFolder**-Recht beachtet.
5. Starten Sie den Dienst **Composer.Core** neu, damit Ihre Änderungen wirksam werden.

# Konfiguration der Windows-Dienste

## Konfigurieren der Wartezeit eines Windows-Dienstes

Beim Starten oder Stoppen eines Windows-Dienstes in der Windows Diensteverwaltung (SCM) oder den Befehlszeilen `net start/ net stop` wartet das Betriebssystem eine bestimmte Zeitspanne, damit die Aktion abgeschlossen werden kann.

Wenn der Dienst nicht innerhalb dieser eingestellten Zeit startet oder stoppt, beendet die Diensteverwaltung den Vorgang mit einem Fehler.

Abhängig vom Prozess kann das Starten oder Stoppen des Prozesses als Windows-Dienst länger dauern als das Standard-Timeout des Betriebssystems dies zulässt.

1. Um den Standard-Timeout zum Starten eines Dienstes zu verlängern, führen Sie die folgenden Schritte aus.
    1. Öffnen Sie die entsprechende Konfigurationsdatei.
    2. Fügen Sie im Element `<appSettings>` die folgende Zeile hinzu:

        ```
        <add key="AdditionalTimeOnStart" value="10000" />
        ```

    3. Ändern Sie das Attribut `value` auf die zusätzlich benötigte Zeitspanne in Millisekunden.
2. Um den Standard-Timeout zum Stoppen eines Dienstes zu verlängern, führen Sie die folgenden Schritte aus.
    1. Öffnen Sie die entsprechende Konfigurationsdatei.
    2. Fügen Sie im Element `<appSettings>` die folgende Zeile hinzu:

        ```
        <add key="AdditionalTimeOnStop" value="10000" />
        ```

    3. Ändern Sie das Attribut `value` auf die zusätzlich benötigte Zeitspanne in Millisekunden.

    **Beispiel**

    Im folgenden Beispiel wird die Zeit zum Starten und Stoppen um jeweils 10 Sekunden verlängert.

    ```
    <configuration>
    …
      <appSettings>
        <add key="AdditionalTimeOnStart" value="10000" />
        <add key="AdditionalTimeOnStop" value="10000" />
      </appSettings>
    …
    </configuration>
    ```

# Netzlaufwerke

## Festlegen des Pfades zu einem Netzlaufwerk

Wenn ein Prozess auf ein Netzlaufwerk zugreifen muss, können Sie den Pfad angeben. Führen Sie die folgenden Schritte aus.

1. Verwenden eines UNC-Pfades

   **Beispiel** `\\NetData\odin`

   Für diese Variante ist keine besondere Konfiguration nötig.

2. Verwenden eines Laufwerksbuchstabens

   Bei dieser Option müssen Sie die erforderliche Laufwerkszuordnung in der entsprechenden Dienst-Konfigurationsdatei definieren.

   Weitere Informationen finden Sie unter Verwenden von Laufwerksbuchstaben für Netzlaufwerke.

## Informationen zu Laufwerkszuordnungen

Laufwerkszuordnungen gelten für ein bestimmtes Windows-Benutzerkonto.

Wird ein Windows-Dienst mit einem Windows-Konto ausgeführt, erwartet das Betriebssystem dass der Dienst die Laufwerksbuchstaben des verwendeten Kontos kennt.

Mit `net use` verbundenen Laufwerke stehen jedoch nur in der interaktiven Windows-Session des Windows-Kontos zur Verfügung..

Wird ein Windows-Dienst gestartet, wird eine Login-Session für diesen Windows-Account erzeugt. Dies ist aber keine interaktive Session, sondern eine spezielle Login-Session zum Ausführen von Diensten. Daher stehen die verbundenen Laufwerke nicht zur Verfügung.

Content Composer-Dienstanwendungen können trotz dieser Einschränkung Netzlaufwerksbuchstaben verwenden.

Enthält die Konfigurationsdatei eines Content Composer Diensts Informationen zur Zuweisung eines Laufwerks zu einem Netzlaufwerk, kann der Dienst ein Verbindung herstellen.

Wenn auf das Netzlaufwerk nicht zugegriffen werden kann, startet der Dienst nicht. Prüfen Sie die Logdatei und das Eventlog auf Informationen zu dieser Aktion.

Wenn ein Dienst beendet wird, wird die Verbindung zum Netzlaufwerk geschlossen.

## Zuordnen eines Laufwerksbuchstabens zu einem Netzlaufwerk

Um einen Laufwerksbuchstaben einem Netzlaufwerk zuzuordnen, führen Sie die folgenden Schritte aus.

1. Öffnen Sie die entsprechende Konfigurationsdatei.
2. Fügen Sie innerhalb des Elements `<configSections>` die folgende Zeile hinzu.

```
<section name="driveMapping"
type="ComposerSuite.Runtime.SystemFramework.DriveMappingConfiguration
,ComposerSuite.Runtime.SystemFramework"/>
```

3. Fügen Sie unterhalb der Zeile `</configSections>` die folgenden Zeilen hinzu:

```
<driveMapping>
   <drives>
      <drive localName="y:" remoteName="\\NetData\mws" />
   </drives>
</driveMapping>
```

4. Ändern Sie `y:` in den benötigten Laufwerksbuchstaben.
5. Ändern Sie `\\NetData\mws` in das benötigte Netzlaufwerk.
6. Optional. Um weitere Laufwerke zuzuweisen, duplizieren Sie die folgende Zeile und passen Sie die Einstellungen an.

```
<drive localName="y:" remoteName="\\NetData\mws" />
```

## Konfiguration der MWS-Dienste

### Das Element *appSettings* in der Composer.MWS.exe.config

```
<appSettings>
   <add key="wordtemplate" value="%Composerdir%Normal.dotm" />
   <add key="DocFilter_LimitsPageCount" value="2000"/>
   <add key="rws_config" value="repository.config" />
   <add key="odinparameterfile" value="odinparameter.config" />
   <add key="disableDebugNamedPipe" value="1" />
   <add key="rws" value="http://localhost:8010/rws" />
   <add key="ows" value="http://localhost:8010/ows/owsrepository" />
   <add key="sts" value="http://localhost:8000/sts" />
   <add key="login" value="http://localhost:8010/mur/login" />
   <add key="mur" value="http://localhost:8010/mur/data" />
   <add key="license" value="http://localhost:8010/mur/license" />
   <add key="Authenticate" value="0" />
   <add key="manVarResultXmlElementType" value="manvar2" />
   <!-- listen to object change notifications so outdated objects are removed
from internal runtime cache -->
   <add key="rwsMulticastIP" value="224.100.0.1" />
   <add key="rwsMulticastPort" value="9050" />
   <add key="rwsMulticastTTL" value="50" />
</appSettings>
```

#### Keys

| Key | Beschreibung |
|---|---|
| wordtemplate | Weitere Informationen finden Sie unter Festlegen der Datei Normal.dotm. |

| Key | Beschreibung |
|---|---|
| DocFilter_LimitsPageCount | Optionaler Parameter, der sich auf die Dokumentenvorschau des Web Clients auswirkt.<br><br>Der numerische Wert gibt die maximale Anzahl der Seiten eines Dokuments an, die in der Vorschau angezeigt werden sollen.<br><br>Wenn die Gesamtzahl der Seiten im Dokument den angegebenen Wert überschreitet, wird ein Fehler angezeigt.<br><br>**Beispiel**: Wenn der Wert auf `2000` gesetzt wurde und das Dokument mehr als 2000 Seiten enthält, tritt bei der Vorschau ein Fehler auf.<br><br>**Defaultwert**: 1000<br><br>**Maximalwert** 4000 |
| rws_config | Nur zur internen Verwendung. |
| odinparameterfile | Nur zur internen Verwendung. |
| disableDebugNamedPipe | Nur zur internen Verwendung. |
| rws | Nur zur internen Verwendung. |
| ows | Nur zur internen Verwendung. |
| sts | Nur zur internen Verwendung. |
| login | Nur zur internen Verwendung. |
| mur | Nur zur internen Verwendung. |
| license | Nur zur internen Verwendung. |
| Authenticate | Nur zur internen Verwendung. |
| manVarResultXmlElementType | Weitere Informationen finden Sie unter Festlegen des manuellen Variablentyps. |

## Festlegen der Normal.dotm

Um festzulegen, welche Normal.dotm der Content Composer Server zum Generieren der Dokumente verwendet, führen Sie die folgenden Schritte aus:

Weitere Informationen finden Sie unter "Word-Dokumentvorlage zuweisen" "Über Word-Dokumentvorlagen" in der *Content Composer Studio Hilfe*.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.MWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die Zeile, die wie folgt beginnt:

   ```
   <add key="wordtemplate"
   ```

3. Setzen Sie das Attribut `value` auf den gewünschten Pfad und Dateinamen.

   **Beispiel**

   ```
   <add key="useInternalDocumentViewer" value="C:\Templates
   \Normal.dotm" />
   ```

   **Hinweis**: Der Defaultwert ist `%Composerdir%Normal.dotm`.

4. Speichern und schließen Sie die Datei.

## Festlegen des manuellen Variablentyps

Um den Elementtyp für manuelle Variablen festzulegen, der in der XML-Datei zurückgegeben wird, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.MWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die Zeile, die wie folgt beginnt:

   ```
   <add key="manVarResultXmlElementType"
   ```

3. Führen Sie einen der folgenden Teilschritte durch.
   - Um nur den Typ `manvar` in die Info-XML aufzunehmen, setzen Sie das Attribut `value` auf `manvar`.
   - Um nur den Typ `manvar2` in die Info-XML aufzunehmen, setzen Sie das Attribut `value` auf `manvar2`.
   - Um beide Typen in die Info-XML aufzunehmen, setzen Sie das Attribut `value` auf `both`.

   **Beispiel**

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

   **Hinweis**: Der Defaultwert ist `both`.

4. Speichern und schließen Sie die Datei.

### Das mws Element in der Composer.MWS.exe.config

```
<mws systemoid="[SYS-OID]" dbalias="MWS" process="MWS_Standard" client_
runtime_mode="false" protect_document="false">
  <serviceusers>
    <serviceuser username="SampleServiceUserName" />
  </serviceusers>
</mws>
```

## Attribute des Elements mws

| Attribut | Beschreibung |
|---|---|
| systemoid | Objekt-ID des Systems, in dem der MWS-Prozess sowie der Datenbank-Alias gespeichert sind. |
| dbalias | Der Name des Datenbank-Alias, der für den Zugriff auf die MWS-Datenbank verwendet wird. |
| process | Der Name des MWS-Prozesses, der sich in dem im Attribut **systemoid** angegebenen System befinden muss. |
| client_runtime_ mode | Nur zur internen Verwendung. |
| protect_ document | Weitere Informationen finden Sie unter Konfigurieren des Dokumentenschutzes |
| set_ processtitle_ using_poolvar | Weitere Informationen finden Sie unter Verwenden einer Pool-Variablen als Prozesstitel |
| result_error_ xml_path | Dieses Attribut wird nur ausgewertet, wenn das Attribut **client_runtime_mode** den Wert FALSE hat.<br>Wird hier ein Pfad angegeben, schreibt Content Composer für jeden Prozess-Aufruf, dessen ResultCode ungleich 0 ist, eine Textdatei in den angegebenen Pfad, die das ResultXml enthält.<br>Der Name der Textdatei ist wie folgt aufgebaut:<br>`[processId]_[yyyyMMdd_HHmmssffffff].txt`<br>**Beispiel**<br>`29111f09-e33f-4782-b071-88a665d23b67_20110922_ 1522384641250.txt` |

| Attribut | Beschreibung |
|---|---|
| | **Beispiel-Konfiguration**<br>• `<mws systemoid="mws" dbalias="MWS" process="MWS_ Standard" client_runtime_mode="FALSE" result_error_xml_ path="\\ServerName\ShareName\FolderName> </mws>`<br>• `<mws systemoid="mws" dbalias="MWS" process="MWS_ Standard" client_runtime_mode="FALSE" result_error_xml_path="c:\temp\mws"> </mws>`<br>• `<mws systemoid="mws" dbalias="MWS" process="MWS_ Standard" client_runtime_mode="FALSE" result_error_xml_path="%MyResultErrorXmlFolder%"> </mws>`<br>Ist dieses Attribut nicht angegeben, wird ein Leerstring verwendet.<br>Falls beim Service-Start der angegebene Pfad nicht gefunden wurde und auch nicht angelegt werden konnte, kann der MWSServiceHost nicht gestartet werden und folgende Fehlermeldung tritt auf:<br>`"runtime_ process";Error;0;;"ModusSuite.MWS.Types.MwsException: Category: <mws>, Token: <ResultErrorXmlPathNotFound> Message: Die Datei "\\ServerName\ShareName\FolderName" konnte nicht gefunden werden. Der angegebene Pfad wurde nicht gefunden. Bitte überprüfen sie die Einstellung in der Konfigurationsdatei für das Attribut 'result_error_xml_path' des Elementes 'mws'`<br>Falls der angegebene Pfad zur Laufzeit nicht mehr verfügbar ist, wird eine entsprechende Fehlermeldung erzeugt, die auch den Inhalt der Datei, die nicht geschrieben werden konnte, enthält. |
| throw_error_ on_missing_ content_control | Weitere Informationen finden Sie unter Handhaben fehlender Content Controls |

## Attribute des Elements serviceuser

| Attribut | Beschreibung |
|---|---|
| username | Das `serviceuser` Element stellt eine Whitelist der Benutzer bereit, die die `Login` Funktion oder die `POST session` URL aus einer Backend-Anwendung aufrufen dürfen, um eine neue Kommunikation oder ein Bündel mit einem Content Composer-Benutzer zu |

| Attribut | Beschreibung |
|---|---|
| | erstellen.<br><br>Aus Sicherheitsgründen können nur Benutzer, die in dieser Whitelist enthalten sind, diese Art der Anmeldung ausführen.<br><br>**Das Ändern dieses Attributs ist nur in folgenden Fällen erforderlich:**<br><br>• Sie rufen die SOAP-Funktion `Login` oder die REST API URL `POST session` von einer Backend-Anwendung aus auf.<br><br>**und**<br><br>• Sie übergeben in der `Login XML` einen Alias-Benutzernamen, der beim Aufruf der `Login` Funktion dem Parameter `options` zugewiesen ist.<br><br>**Beispiel**<br><br>Die Backend-Anwendung läuft unter dem Benutzernamen `mws-service-user_1`.<br><br>Sie übergeben den Benutzernamen `mws-service-user_1` für den SOAP-Parameter `username` oder im REST Authorization Header.<br><br>Als Alias-Benutzername soll im SOAP-Parameter `alias` oder REST-Parameter `aliasUsername` ein Content Composer-Benutzer übergeben werden, den die Sitzung beim Erstellen einer neuen Kommunikation oder eines Bündels verwendet.<br><br>Um diese Art der Anmeldung zuzulassen, müssen Sie den Benutzernamen `mws-service-user_1` zur Whitelist der `serviceusers` hinzufügen. Weitere Informationen finden Sie im Kapitel Definieren von Service-Usern.<br><br>Weitere Informationen finden Sie unter "Login" im *Content Composer Web Services Handbuch*. |

## Konfigurieren des Dokumentenschutzes

Um zu konfigurieren, ob der Server den Dokumentenschutz direkt nach der Dokumentenerstellung aktiviert, führen Sie die folgenden Schritte aus.

Dieses Attribut gilt für Dokumente mit Passwortschutz und wird nur ausgewertet, wenn `client_runtime_mode` auf `False` gesetzt ist.

**Hinweis**: Der Dokumentenschutz wird nicht angewendet, wenn ein Dokument in der Vorschau oder in **OnBase** angezeigt wird.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.MWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die Zeile, die wie folgt beginnt:

   ```
   <mws systemoid=
   ```

3. Führen Sie einen der folgenden Teilschritte durch.

- Um den Dokumentenschutz nach der Dokumentenerstellung auf dem Server zu aktivieren, setzen Sie das Attribut `protect_document` auf `True`.
- Um den Dokumentschutz nur auf dem Client zu aktivieren, setzen Sie das Attribut `protect_document` auf `False`.

**Beispiel**

```
<mws systemoid="dm" dbalias="odin_mws" process="MWS_Standard" client_
runtime_mode="FALSE" protect_document="FALSE" />
```

**Hinweis**: Die Default-Einstellung ist `False`.

4. Speichern und schließen Sie die Datei.

## Definieren von Service-Usern

Um einen oder mehrere Service-User anzugeben, führen Sie die folgenden Schritte aus:Um die Dienstadressen anzugeben, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.MWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie nach dem Element `serviceusers` innerhalb des Elements `mws`.
3. Ändern Sie das Attribut `username` auf den Namen des Benutzers, der die Backend-Anwendung ausführt, welche die `Login`-Funktion aufruft.

   **Beispiel**

   ```
   <mws systemoid="[SYS-OID]" dbalias="MWS" process="MWS_Standard"
   client_runtime_mode="false" protect_document="false">
      <serviceusers>
         <serviceuser username="mws-service-user-1" />
      </serviceusers>
     </mws>
   ```

   **Hinweis**: Möglicherweise müssen Sie das `serviceusers`-Element einfügen, wenn es noch nicht vorhanden ist.

4. Optional. Um weitere Benutzer hinzuzufügen, kopieren Sie das `serviceuser` Element.

   **Beispiel**

   ```
   <serviceusers>
         <serviceuser username="mws-service-user-1" />
         <serviceuser username="mws-service-user-2" />
      </serviceusers>
   ```

5. Speichern und schließen Sie die Datei.

## Verwenden einer Pool-Variablen als Prozesstitel

Um den Inhalt einer SYSTEMPOOL-Variablen als Prozesstitel zu verwenden wenn ein Prozess geschlossen oder weitergeleitet wird, führen Sie die folgenden Schritte aus:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.MWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die Zeile, die wie folgt beginnt:

   ```
   <mws systemoid=
   ```

3. Fügen Sie das Attribut `set_processtitle_using_poolvar` innerhalb des `mws` Elements hinzu.

   **Beispiel**

   ```
   <mws systemoid="TestSystem" dbalias="MWS" process="MWS_Standard"
   client_runtime_mode="false" protect_document="false" set_processtitle_
   using_poolvar="ProcessTitlePoolVar">
   ```

   **Hinweis**: Die Default-Einstellung ist ein Leerstring.

4. Speichern und schließen Sie die Datei.

## Handhaben fehlender Content Controls

Um das Verhalten der Anwendung im Falle eines fehlenden Content Controls zu konfigurieren, führen Sie die folgenden Schritte aus:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.MWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die Zeile, die wie folgt beginnt:

   ```
   <mws systemoid=
   ```

3. Fügen Sie das Attribut `throw_error_on_missing_content_control` innerhalb des `mws` Elements hinzu.
   - Um die Dokumentengenerierung mit einer Fehlermeldung zu beenden, wenn ein Content Control einer Textblockvariablen fehlt, setzen Sie den Wert auf `True`.
   - Um eine Warnmeldung zu protokollieren und die Dokumentgenerierung fortzusetzen, wenn ein Content Control einer Textblockvariablen fehlt, setzen Sie den Wert auf `False`.

   **Beispiel**

   ```
   <mws systemoid="TestSystem" dbalias="MWS" process="MWS_Standard"
   client_runtime_mode="False" protect_document="False" throw_error_on_
   missing_content_control=True>
   ```

   **Hinweis**: Die Default-Einstellung ist `True`.

4. Speichern und schließen Sie die Datei.

## Ausführen des Composer.MWS-Dienstes ohne Administratorrechte

So können Sie den Composer.MWS-Dienst mit einem Benutzerkonto ohne Administratorrechte ausführen:

1. Um die URLs zu reservieren, müssen Sie die folgenden Befehle in einem **CMD-Fenster** mit Administratorrechten ausführen. Ersetzen Sie hierzu die Portnummer und den Kontonamen durch die entsprechenden Werte. Wenn sie http verwenden,ersetzen Sie `url=https` durch `url=http`.
    1. `netsh http add urlacl` ***url=https***`://+:`***[Port Number]***`/mws/ user=`***[Windows-user-account]***
    2. `netsh http add urlacl` ***url=https***`://+:`***[Port Number]***`/mws/ user=`***[Windows-user-account]***
    3. `netsh http add urlacl` ***url=https***`://+:`***[Port Number]***`/mwsbasic/ user=`***[Windows-user-account]***
    4. `netsh http add urlacl` ***url=https***`://+:`***[Port Number]***`/mwsrest/ user=`***[Windows-user-account]***
    5. `netsh http add urlacl` ***url=https***`://+:`***[Port Number]***`/RemoteDebugger/ user=`***[Windows-user-account]***
    6. `netsh http add urlacl` ***url=https***`://+:`***[Port Number]***`/RemoteControlService/ user=`***[Windows-user-account]***
    7. `netsh http add urlacl url=https://+:`***[Port Number]*** `user=`***[Windows-user-account]***
2. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.MWS.exe.config** mit einem Texteditor.
3. Für den ersten Befehl suchen Sie nach dem **ersten**`<add baseAddress`-Element, das mit `/mws` endet, und verwenden die in diesem Element verwendete Portnummer, um den Befehl auszuführen.

    **Beispiel**: `<add baseAddress="http://localhost:`***8010***`/mws" />`

    **Auszuführender Befehl**: `netsh http add urlacl url=https://+:8010/mws/ user=`***[Windows-user-account]***
4. Für den zweiten Befehl suchen Sie nach dem **zweiten**`<add baseAddress`-Element, das mit `/mws` endet, und verwenden die in diesem Element verwendete Portnummer, um den Befehl auszuführen.

    **Beispiel**: `<add baseAddress="http://localhost:`***8011***`/mws" />`

    **Auszuführender Befehl**: `netsh http add urlacl url=https://+:8011/mws/ user=`***[Windows-user-account]***
5. Für den dritten Befehl suchen Sie nach dem `<add baseAddress`-Element, das mit `/mwsbasic` endet, und verwenden die in diesem Element verwendete Portnummer, um den Befehl auszuführen.

    **Beispiel**: `<add baseAddress="https://localhost:8111/mwsbasic" />`

    **Auszuführender Befehl**: `netsh http add urlacl url=https://+:8111/mwsbasic/ user=`***[Windows-user-account]***

6. Für den vierten Befehl suchen Sie nach dem `<add baseAddress`-Element, das mit `/mwsrest` endet, und verwenden die in diesem Element verwendete Portnummer, um den Befehl auszuführen.

   **Beispiel**: `<add baseAddress="https://localhost:8111/mwsrest" />`

   **Auszuführender Befehl**: `netsh http add urlacl url=https://+:8111/mwsrest/ user=`***`[Windows-user-account]`***

7. Für den fünften Befehl suchen Sie nach dem `<endpoint address`-Element für `/RemoteDebugger`, und verwenden die in diesem Element verwendete Portnummer, um den Befehl auszuführen.

   **Beispiel**: `<endpoint address="http://127.0.0.1:8012/RemoteDebugger" ...`

   **Auszuführender Befehl**: `netsh http add urlacl url= https://+:8012/RemoteDebugger/ user=`***`[Windows-user-account]`***

8. Für den sechsten Befehl suchen Sie nach dem `<endpoint address`-Element für `/RemoteControlService`, und verwenden die in diesem Element verwendete Portnummer, um den Befehl auszuführen.

   **Beispiel**: `<endpoint address="http://127.0.0.1:4721/RemoteControlService" ...`

   **Auszuführender Befehl**: `netsh http add urlacl url= https://+:4721/RemoteControlService/ user=`***`[Windows-user-account]`***

9. Für den siebten Befehl suchen Sie nach der Zeile die mit `<mwswebapi ipport=` begint und verwenden die in diesem Element verwendete Portnummer, um den Befehl auszuführen.

   **Beispiel**: `<mwswebapi ipport="`***`9010`***`" ...`

   **Auszuführender Befehl**: `netsh http add urlacl url=https://+:9010/ user=` ***`[Windows-user-account]`***

10. Führen Sie den folgenden Befehl aus und verwenden Sie dabei alle in den zuvor ausgeführten Befehlen verwendeten Ports.

    **Beispiel**: `netsh http add urlacl url=https://+:`***`[PortNumber]`***`/ user=` ***`[Windows-user-account]`***

    **Auszuführende Befehle:**

    ```
    netsh http add urlacl url=https://+:8010/ user=[Windows-user-account]
    netsh http add urlacl url=https://+:8011/ user=[Windows-user-account]
    netsh http add urlacl url=https://+:8111/ user=[Windows-user-account]
    netsh http add urlacl url=https://+:8012/ user=[Windows-user-account]
    netsh http add urlacl url=https://+:4721/ user=[Windows-user-account]
    ```

11. Öffnen Sie auf dem Content Composer Server, den Zertifikatspeicher **Local Computer - Personal - Certificates**.
12. Rechtsklicken Sie auf das Zertifikat **composer.server** und klicken Sie dann auf **All Tasks | Manage Private Keys**.

13. Fügen Sie im Dialog **Permission for composer.server private keys** das in diesem Kapitel verwendete Benutzerkonto hinzu.
14. Vergewissern Sie sich, dass das hinzugefügte Benutzerkonto mindestens Lesezugriff hat, und klicken Sie dann auf **OK**.
15. Rechtsklicken Sie auf das Zertifikat **composer.service** und klicken Sie dann auf **All Tasks | Manage Private Keys**.
16. Fügen Sie im Dialog **Permission for composer.service private keys** das in diesem Kapitel verwendete Benutzerkonto hinzu.
17. Vergewissern Sie sich, dass das hinzugefügte Benutzerkonto mindestens Lesezugriff hat, und klicken Sie dann auf **OK**.
18. Führen Sie für das in diesem Kapitel verwendete Benutzerkonto die Schritte unter **Vergeben von Zugriffsrechten für den RSA-Container** in der Content Composer Installationsanleitung aus
19. Wenn Sie `HTTPS` verwenden, müssen Sie dem Benutzerkonto die erforderlichen Rechte für das SSL-Zertifikat erteilen. Führen Sie die folgenden Teilschritte aus:
    1. Öffnen Sie auf dem Content Composer-Server den Certificate Manager und suchen Sie das SSL-Zertifikat.
    2. Rechtsklicken Sie auf das SSL-Zertifikat und klicken Sie dann auf **All Tasks | Manage Private Keys**.
    3. Fügen Sie im Dialog **Permission** das in diesem Kapitel verwendete Benutzerkonto hinzu.
    4. Vergewissern Sie sich, dass das hinzugefügte Benutzerkonto mindestens Lesezugriff hat, und klicken Sie dann auf **OK**.

# Konfiguration der Kommunikation

## Informationen zur Content Composer Kommunikation

Content Composer besteht aus verschiedenen Server-Diensten und Clients.

Diese Client- und Server-Dienste kommunizieren über verschiedene SOAP Web-Service Schnittstellen, die auf Basis der Microsoft Windows Communication Foundation (kurz: WCF) zur Verfügung gestellt werden.

Damit die Clients und Server miteinander kommunizieren können, müssen die Clients die Adressen der Server-Dienste im Netzwerk kennen.

Da die Kommunikation über das HTTP-Protokoll erfolgt, besitzt eine Server-Adresse den typischen Aufbau einer Webpage-Adresse:

**Beispiel**

```
http://Hostname:Port/sts
```

Im Folgenden werden für die einzelnen Web-Service Schnittstellen die zuständigen Konfigurationseinstellungen jeweils für Client und Server aufgeführt.

**Hinweis**: Damit die Kommunikation funktioniert, muss in den Client- und Server-Einstellungen jeweils derselbe Netzwerkname und Port eingetragen sein.

# HTTPS-Konfiguration für den WCF-Kommunikationskanal

Die Standardinstallation von Content Composer verwendet das HTTP-Protokoll für die Kommunikation zwischen Content Composer Windows Anwendungen und dem Content Composer Windows Server Endpunkt.

Um den WCF-Kommunikationskanal für die Verwendung von HTTPS zu konfigurieren, folgen Sie den Anweisungen in diesem Kapitel.

**Voraussetzungen**

- Ein digitales SSL-Zertifikat, das Sie entweder erwerben oder als selbstsigniertes Zertifikat erstellen können.

  Das Zertifikat muss die folgenden Subject Alternative Names enthalten:

  - Den DNS-Namen `localhost`.

    Dies ist erforderlich, wenn die Content Composer Serverkonfigurationsdateien `localhost` in den im Element `<appSettings>` angegebenen URLs verwenden.

  - Den DNS-Namen des Content Composer Servers.

    Wir empfehlen, ein SSL-Zertifikat zu erstellen, das die DNS-Namen aller Content Composer Server enthält, wenn Ihre Installation mehrere Content Composer-Server umfasst, die hinter einem Load Balancer laufen.

  - Der DNS-Name des Load Balancers, wenn Ihre Installation mehrere Content Composer Server umfasst, die hinter einem Load Balancer laufen.
- Eine vollständig konfigurierte Content Composer Installation.

## Installieren des SSL-Zertifikats

Um das Zertifikat zu installieren, führen Sie die folgenden Schritte auf allen Content Composer-Servern, Content Composer Studio-Workstations und Content Composer Windows Client-Workstations durch.

1. Importieren Sie das SSL-Zertifikat in die folgenden Windows-Zertifikatspeicher:
   - Eigene Zertifikate
   - Vertrauenswürdige Stammzertifizierungsstellen
2. Um das Zertifikat an die in den Content Composer-Serverkonfigurationsdateien verwendete IP-Portnummer zu binden, führen Sie auf jedem Content Composer-Server die folgenden Teilschritte aus.
   1. Öffnen Sie die Windows Eingabeaufforderung als Administrator.
   2. Führen Sie den folgenden Befehl aus und ersetzen Sie dabei den Wert des Parameters `certhash` durch den Thumbprint-Wert Ihres Zertifikats.

      **Hinweis**: Die Änderung des IP-Ports im Parameter `ipport` ist optional.

      ```
      netsh http add sslcert ipport=0.0.0.0:8100 certhash=
      [Thumbprint] appid={[any GUID]}
      ```

## Aktivieren von HTTPS für den WCF-Kommunikationskanal

Um HTTPS zu aktivieren, führen Sie die folgenden Schritte auf jedem Content Composer Server, jeder Content Composer Studio Workstation und jeder Content Composer Windows Client Workstation aus.

1. Kopieren Sie aus dem Verzeichnis **%Composerdir%** die folgenden Dateien in ein Backup-Verzeichnis.

   **Hinweis**: Abhängig davon, ob Sie eine Content Composer Server-, Content Composer Studio- oder Content Composer Client-Installation aktualisieren, sind möglicherweise nicht alle CONFIG-Dateien vorhanden.

   - Composer.Core.exe.config
   - Composer.MWS.exe.config
   - Composer.OWS.exe.config
   - Composer.XWS.exe.config
   - Composer.Studio.exe.config
   - Composer.Client.exe.config
   - Composer.ConsoleHost.exe.config
   - Composer.WindowsServiceHost.exe.config
   - MonalisaEngine.exe.config

2. Wenn Sie eine Content Composer Studio oder Content Composer Windows Client Installation aktualisieren, führen Sie den folgenden Teilschritt aus.

   - Kopieren Sie auf Content Composer Server aus dem Verzeichnis **%Composerdir%admin\configuration** die Datei **HTTPSProtocol.ps1** in das Verzeichnis **%Composerdir%admin\configuration** auf der Content Composer Studio oder Client Workstation.

     **Hinweis**: Möglicherweise müssen Sie dieses Verzeichnis erstellen, wenn es noch nicht vorhanden ist.

3. Öffnen Sie als Administrator ein CMD-Fenster und wechseln Sie in das Verzeichnis **%Composerdir%admin\configuration**.
4. Führen Sie einen der folgenden Teilschritte aus und verwenden Sie dabei die gleiche Portnummer, die Sie für die Installation des SSL-Zertifikats verwendet haben:

   - Um HTTPS für alle WCF-Kommunikationskanäle an IP-Port `8100` zu aktivieren, führen Sie den folgenden Befehl aus:

     ```
     powershell -file Set-HTTPSProtocol.ps1
     ```

   - Um HTTPS für alle WCF-Kommunikationskanäle an einem IP-Port Ihrer Wahl zu aktivieren, führen Sie den folgenden Befehl aus, wobei Sie `[IPPort]` durch die gewünschte Portnummer ersetzen:

     ```
     powershell -file Set-HTTPSProtocol.ps1 -ipport [IPPort]
     ```

     **Beispiel**

     ```
     powershell -file Set-HTTPSProtocol.ps1 -ipport 8500
     ```

Wenn keine Fehler aufgetreten sind, finden Sie die geänderten Dateien im Verzeichnis **%Composerdir%targets**.

5. Überschreiben Sie die im Verzeichnis **%Composerdir%** vorhandenen CONFIG-Dateien mit ihren Entsprechungen aus dem Verzeichnis **%Composerdir%targets**.
6. Starten Sie auf dem Content Composer-Server alle Content Composer-Dienste neu.

## Importieren der RSA Encryption Keys auf dem Client

Nachdem Sie die HTTPS-Konfiguration für den WCF-Kommunikationskanal aktiviert haben, müssen Sie die mit der ersten Content Composer Serverinstallation erstellten Encryption Keys auf jeden Content Composer-Client-Arbeitsplatz importieren. Führen Sie die folgenden Schritte aus:

1. Kopieren Sie die folgenden Dateien aus Ihrem Backup-Speicher in das **%Composerdir%**-Verzeichnis:
   - encryption_keys.config
   - CoCoEncryptionRSAKey.xml

   **Hinweis**: Informationen zum Backup-Speicher finden Sie unter "Exportieren und Sichern der Encryption Keys" in der *Content Composer Installationsanleitung*.
2. Öffnen Sie als Administrator ein CMD-Fenster und wechseln Sie in das Verzeichnis **%Composerdir%**.
3. Um den Encryption-Key zu importieren, führen Sie folgenden Befehl aus:

   ```
   Composer.EncryptionTool importkey
   ```
4. Löschen Sie aus Sicherheitsgründen nach dem Importieren die folgende Datei aus dem Verzeichnis **%Composerdir%**:
   - CoCoEncryptionRSAKey.xml
5. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
6. Fügen Sie die folgende Zeile innerhalb des `<appsettings>` Elements ein:

   ```
   <add key="ProtectedConfiguration" value="encryption_keys.config" />
   ```
7. Speichern und schließen Sie die Datei.

## Vergeben von Zugriffsrechten für den RSA-Container

Um jedem Windows-Konto, das einen Content Composer Client ausführt, Zugriffsrechte auf den RSA-Container zu gewähren, führen Sie die folgenden Schritte aus:

1. Öffnen Sie ein CMD-Fenster als Administrator und wechseln Sie in das Verzeichnis **%windir%\Microsoft.NET\Framework64\v4.0.30319**.
2. Führen Sie den folgenden Befehl aus, der dem Benutzer `CindySmith`, der Mitglied der Domäne `OnBase` ist, Zugriffsrechte gewährt. Ersetzen Sie zuvor Domäne und Benutzername durch die erforderlichen Werte.

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

3. Wiederholen Sie den vorherigen Schritt für jedes Windows-Konto, unter dem ein Content Composer Client ausgeführt wird.

### Problembehebung

- Um vorhandene Port-Bindings zu überprüfen, führen Sie die folgenden Teilschritte durch.
  - Öffnen Sie die Windows Eingabeaufforderung als Administrator.
  - Führen Sie folgenden Befehl aus:

    ```
    netsh http show sslcert
    ```

- Um eine vorhandene Port-Binding zu entfernen, führen Sie die folgenden Teilschritte durch.
  - Öffnen Sie die Windows Eingabeaufforderung als Administrator.
  - Führen Sie folgenden Befehl aus und ersetzen Sie dabei `[IPPort]` durch die zu entfernende Portnummer:

    ```
    netsh http delete sslcert ipport=0.0.0.0:[IPPort]
    ```

## Security Token Service (sts)

### Konfigurieren des Security Token Services (sts) auf dem Server

Die spezifischen Funktionen dieses Dienstes werden durch den Dienst **Composer.Core** bereitgestellt.

Um die individuellen Einstellungen für diesen Server-Endpunkt zu konfigurieren, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Core.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Zeile.

   ```
   <add baseAddress ="http://localhost:8000/sts" />
   ```

3. Prüfen Sie das Attribut `baseAddress` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als value muss dabei die host-base-Addresse des Services angegeben werden.

   **Beispiel**

   ```
   <add baseAddress ="http://Hyland:8000/sts" />
   ```

### Konfigurieren des Security Token Services (sts) auf dem Client

Um die Dienstadresse anzugeben, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Zeile.

   ```
   <add key="sts" value="http://localhost:8000/sts" />
   ```

3. Prüfen Sie das Attribut `value` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als value muss dabei die host-base-Addresse des Services angegeben werden.

**Beispiel**

Im folgenden Beispiel kommuniziert der Client mit dem Server-Dienst auf dem Computer mit dem Netzwerknamen `Hyland` am IP-Port `8000`:

```
<add key="sts" value="http://Hyland:8000/sts" />
```

## Composer User Repository (mur)

### Konfigurieren des Composer User Repositorys (mur) auf dem Server

Die spezifischen Funktionen dieses Dienstes werden durch den Dienst **Composer.Core** bereitgestellt.

Um die individuellen Einstellungen für diesen Server-Endpunkt zu konfigurieren, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Core.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Zeile.

   ```
   <add baseAddress ="http://localhost:8010/mur" />
   ```

3. Prüfen Sie das Attribut `baseAddress` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als value muss dabei die host-base-Addresse des Services angegeben werden.

   **Beispiel**

   ```
   <add baseAddress ="http://Hyland:8010/mur" />
   ```

### Konfigurieren des Composer User Repositorys (mur) auf dem Client

Um die Dienstadressen anzugeben, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Zeile.

   ```
   <add key="login" value="http://localhost:8010/mur/login" />
   ```

3. Prüfen Sie das Attribut `value` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als value für den Key **login** muss dabei die Login-Adresse angegeben werden. Diese Adresse setzt sich dabei aus der Hostbase-Adresse und der entsprechenden Endpoint-Adresse des Services zusammen.

4. Suchen Sie die folgenden Zeile.

   ```
   <add key="mur" value="http://localhost:8010/mur/data" />
   ```

5. Prüfen Sie das Attribut `value` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als value für den Key **mur** muss die Data-Adresse angegeben werden. Diese Adresse setzt sich dabei aus der Hostbase-Adresse und der entsprechenden Endpoint-Adresse des Services zusammen.

**Beispiel**

Im folgenden Beispiel kommuniziert der Client mit dem Server-Dienst auf dem Computer mit dem Netzwerknamen `Hyland` am IP-Port `8010`:

```
<add key="login" value="http://Hyland:8010/mur/login" />
<add key="mur" value="http://Hyland:8010/mur/data" />
```

## DataProviderService (dpws)

### Konfigurieren des DataProviderServices (dpws) auf dem Server

Die spezifischen Funktionen dieses Dienstes werden durch den Dienst **Composer.Core** bereitgestellt.

Um die individuellen Einstellungen für diesen Server-Endpunkt zu konfigurieren, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Core.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Zeile.

   ```
   <add baseAddress ="http://localhost:8010/dataprovider" />
   ```

3. Prüfen Sie das Attribut `baseAddress` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als value muss dabei die host-base-Addresse des Services angegeben werden.

   **Beispiel**

   ```
   <add baseAddress ="http://Hyland:8010/dataprovider" />
   ```

### Konfigurieren des DataProviderServices (dpws) auf dem Client

Um die Dienstadresse anzugeben, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Zeile.

   ```
   <add key="dpws" value="http://localhost:8010/dataprovider" />
   ```

3. Prüfen Sie das Attribut `value` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als value muss dabei die host-base-Addresse des Services angegeben werden.

   **Beispiel**

   Im folgenden Beispiel kommuniziert der Client mit dem Server-Dienst auf dem Computer mit dem Netzwerknamen `Hyland` am IP-Port `8010`:

   ```
   <add key="dpws" value="http://Hyland:8010/dataprovider" />
   ```

# RepositoryWebService (rws)

## Konfigurieren des RepositoryWebServices (rws) auf dem Server

Die spezifischen Funktionen dieses Dienstes werden durch den Dienst **Composer.Core** bereitgestellt.

Um die individuellen Einstellungen für diesen Server-Endpunkt zu konfigurieren, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Core.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Zeile.

   ```
   <add baseAddress ="http://localhost:8010/rws" />
   ```

3. Prüfen Sie das Attribut `baseAddress` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als value muss dabei die host-base-Addresse des Services angegeben werden.

   **Beispiel**

   ```
   <add baseAddress ="http://Hyland:8010/rws" />
   ```

## Konfigurieren des RepositoryWebServices (rws) auf dem Client

Um die Dienstadresse anzugeben, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Zeile.

   ```
   <add key="rws" value="http://localhost:8010/rws" />
   ```

3. Prüfen Sie das Attribut `value` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als value muss dabei die host-base-Addresse des Services angegeben werden.

   **Beispiel**

   Im folgenden Beispiel kommuniziert der Client mit dem Server-Dienst auf dem Computer mit dem Netzwerknamen `Hyland` am IP-Port `8010`:

   ```
   <add key="rws" value="http://Hyland:8010/rws" />
   ```

# OdinWebService (ows und owsbasic)

## Konfigurieren des OdinWebServices (ows) auf dem Server

Die spezifischen Funktionen dieses Dienstes werden durch den Dienst **Composer.OWS** bereitgestellt. Der Dienst wird als Windows-Dienst als Teil der Content Composer Standardinstallation installiert.

Um die individuellen Einstellungen für diesen Server-Endpunkt zu konfigurieren, führen Sie die folgenden Schritte aus.

1. Kopieren Sie aus dem Verzeichnis **%Composerdir%** die folgenden Dateien in ein Backup-

Verzeichnis. **Hinweis**:

2. Suchen Sie die folgenden Zeile.

   ```
   <add baseAddress ="http://localhost:8010/ows" />
   ```

3. Prüfen Sie das Attribut `baseAddress` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als value muss dabei die host-base-Addresse des Services angegeben werden.

   **Beispiel**

   ```
   <add baseAddress ="http://Hyland:8010/ows" />
   ```

## Konfigurieren des OdinWebServices (ows) auf dem Client

Um die Dienstadresse anzugeben, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Zeile.

   ```
   <add key="ows" value="http://localhost:8010/ows/owsrepository" />
   ```

3. Prüfen Sie das Attribut `value` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als Wert muss dabei die owsrepository-Adresse angegeben werden. Diese Adresse setzt sich dabei aus der Hostbase-Adresse und der entsprechenden Endpoint-Adresse des Services zusammen.

   **Beispiel**

   Im folgenden Beispiel kommuniziert der Client mit dem Server-Dienst auf dem Computer mit dem Netzwerknamen `Hyland` am IP-Port `8010`:

   ```
   <add key="ows" value="http://Hyland:8010/ows/owsrepository" />
   ```

## Client-Konfiguration owsbasic

Die folgende WSDL-URL kann von einem Drittanbieter-Client wie z. B. einem JAVA-Client verwendet werden, um die Servicedefinition abzurufen.

```
http://localhost:8010/owsbasic?wsdl
```

Diese Adresse setzt sich aus der host-base-Adresse gefolgt von ?wsdl zusammen.

**Beispiel**

Im folgenden Beispiel wird die WSDL URL des Server-Dienstes auf dem Computer mit dem Netzwerknamen Hyland am IP-Port 8010 angegeben.

```
http://Hyland:8010/owsbasic?wsdl
```

### Konfigurieren des OdinWebServices (owsbasic) auf dem Server

Die spezifischen Funktionen dieses Dienstes werden durch den Dienst **Composer.OWS** bereitgestellt. Der Dienst wird als Windows-Dienst als Teil der Content Composer Standardinstallation installiert.

Um die individuellen Einstellungen für diesen Server-Endpunkt zu konfigurieren, führen Sie die folgenden Schritte aus.

1. Kopieren Sie aus dem Verzeichnis **%Composerdir%** die folgenden Dateien in ein Backup-Verzeichnis. **Hinweis**:
2. Suchen Sie die folgenden Zeile.

   ```
   <add baseAddress ="http://localhost:8010/owsbasic" />
   ```

3. Prüfen Sie das Attribut `baseAddress` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als value muss dabei die host-base-Addresse des Services angegeben werden.

   **Beispiel**

   ```
   <add baseAddress ="http://Hyland:8010/owsbasic" />
   ```

## XdataWebService (xws und xwsbasic)

### Konfigurieren des XdataWebServices (xws) auf dem Server

Die spezifischen Funktionen dieses Dienstes werden durch den Dienst **Composer.XWS** bereitgestellt. Der Dienst wird als Windows-Dienst als Teil der Content Composer Standardinstallation installiert.

Um die individuellen Einstellungen für diesen Server-Endpunkt zu konfigurieren, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.XWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Zeile.

   ```
   <add baseAddress ="http://localhost:8010/xws" />
   ```

3. Prüfen Sie das Attribut `baseAddress` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als value muss dabei die host-base-Addresse des Services angegeben werden.

   **Beispiel**

   ```
   <add baseAddress ="http://Hyland:8010/xws" />
   ```

### Konfigurieren des XdataWebServices (xws) auf dem Client

Um die Dienstadresse anzugeben, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Zeile.

   ```
   <add key="xws" value=" http://localhost:8010/xws/selection" />
   ```

3. Prüfen Sie das Attribut `value` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als Wert muss dabei die **xws**-Adresse angegeben werden. Diese Adresse setzt sich dabei aus der Hostbase-Adresse und der entsprechenden Endpoint-Adresse des Services zusammen.

   **Beispiel**

   Im folgenden Beispiel kommuniziert der Client mit dem Server-Dienst auf dem Computer mit dem Netzwerknamen `Hyland` am IP-Port `8010`:

   ```
   <add key="xws" value="http://Hyland:8010/xws/selection" />
   ```

## Client-Konfiguration xwsbasic

Die folgende WSDL-URL kann von einem Drittanbieter-Client wie z. B. einem JAVA-Client verwendet werden, um die Servicedefinition abzurufen.

```
http://localhost:8010/xwsbasic?wsdl
```

Diese Adresse setzt sich aus der host-base-Adresse gefolgt von ?wsdl zusammen.

**Beispiel**

Im folgenden Beispiel wird die WSDL URL des Server-Dienstes auf dem Computer mit dem Netzwerknamen `Hyland` am IP-Port `8010` angegeben:

```
http://Hyland:8010/xwsbasic?wsdl
```

## Konfigurieren des XdataWebServices (xwsbasic) auf dem Server

Die spezifischen Funktionen dieses Dienstes werden durch den Dienst **Composer.XWS** bereitgestellt. Der Dienst wird als Windows-Dienst als Teil der Content Composer Standardinstallation installiert.

Um die individuellen Einstellungen für diesen Server-Endpunkt zu konfigurieren, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.XWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Zeile.

   ```
   <add baseAddress ="http://localhost:8010/xwsbasic" />
   ```

3. Prüfen Sie das Attribut `baseAddress` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als value muss dabei die host-base-Addresse des Services angegeben werden.

   **Beispiel**

   ```
   <add baseAddress ="http://Hyland:8010/xwsbasic" />
   ```

# ComposerWebService (mws und mwsbasic)

## Konfigurieren des ComposerWebServices (mws) auf dem Server

Die spezifischen Funktionen dieses Dienstes werden durch den Dienst **Composer.MWS** bereitgestellt. Der Dienst wird als Windows-Dienst als Teil der Content Composer Standardinstallation installiert.

Um die individuellen Einstellungen für diesen Server-Endpunkt zu konfigurieren, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.MWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Zeile.

   ```
   <add baseAddress ="http://localhost:8011/mws" />
   ```

3. Prüfen Sie das Attribut `baseAddress` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als value muss dabei die host-base-Addresse des Services angegeben werden.

   **Beispiel**

   ```
   <add baseAddress ="http://Hyland:8011/mws" />
   ```

## Konfigurieren des ComposerWebServices (mws) auf dem Client

Um die Dienstadresse anzugeben, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Zeile.

   ```
   <add key="mws" value=" http://localhost:8011/mws/mwsprocess" />
   ```

3. Prüfen Sie das Attribut `value` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als Wert muss dabei die **mws**-Adresse angegeben werden. Diese Adresse setzt sich dabei aus der Hostbase-Adresse und der entsprechenden Endpoint-Adresse des Services zusammen.

   **Beispiel**

   Im folgenden Beispiel kommuniziert der Client mit dem Server-Dienst auf dem Computer mit dem Netzwerknamen `Hyland` am IP-Port `8011`:

   ```
   <add key="mws" value="http://Hyland:8011/mws/mwsprocess" />
   ```

## Client-Konfiguration mwsbasic

Die folgende WSDL-URL kann von einem Drittanbieter-Client wie z. B. einem JAVA-Client verwendet werden, um die Servicedefinition abzurufen.

```
http://localhost:8011/mwsbasic?wsdl
```

Diese Adresse setzt sich aus der host-base-Adresse gefolgt von ?wsdl zusammen.

**Beispiel**

Im folgenden Beispiel wird die WSDL URL des Server-Dienstes auf dem Computer mit dem Netzwerknamen `Hyland` am IP-Port `8011` angegeben:

```
http://Hyland:8011/mwsbasic?wsdl
```

### Konfigurieren des ComposerWebServices (mwsbasic) auf dem Server

Die spezifischen Funktionen dieses Dienstes werden durch den Dienst **Composer.MWS** bereitgestellt. Der Dienst wird als Windows-Dienst als Teil der Content Composer Standardinstallation installiert.

Um die individuellen Einstellungen für diesen Server-Endpunkt zu konfigurieren, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.MWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Zeile.

   ```
   <add baseAddress ="http://localhost:8011/mwsbasic" />
   ```

3. Prüfen Sie das Attribut `baseAddress` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als value muss dabei die host-base-Addresse des Services angegeben werden.

   **Beispiel**

   ```
   <add baseAddress ="http://Hyland:8011/mwsbasic" />
   ```

## ComposerRepositoryWebService (mwsrws und mwsrwsbasic)

### Konfigurieren des ComposerRepositoryWebServices (mwsrws) auf dem Server

Die spezifischen Funktionen dieses Dienstes werden durch den Dienst **Composer.MWS** bereitgestellt. Der Dienst wird als Windows-Dienst als Teil der Content Composer Standardinstallation installiert.

Um die individuellen Einstellungen für diesen Server-Endpunkt zu konfigurieren, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.MWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Zeile.

   ```
   <add baseAddress ="http://localhost:8010/mws" />
   ```

3. Prüfen Sie das Attribut `baseAddress` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als value muss dabei die host-base-Addresse des Services angegeben werden.

   **Beispiel**

   ```
   <add baseAddress ="http://Hyland:8010/mws" />
   ```

## Konfigurieren des ComposerRepositoryWebServices (mwsrws) auf dem Client

Um die Dienstadresse anzugeben, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Zeile.

   ```
   <add key="mwsrws" value=" http://localhost:8010/mws/mwsrepository" />
   ```

3. Prüfen Sie das Attribut `value` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als Wert muss dabei die **mwsrepository**-Adresse angegeben werden. Diese Adresse setzt sich dabei aus der Hostbase-Adresse und der entsprechenden Endpoint-Adresse des Services zusammen.

   **Beispiel**

   Im folgenden Beispiel kommuniziert der Client mit dem Server-Dienst auf dem Computer mit dem Netzwerknamen `Hyland` am IP-Port `8010`:

   ```
   <add key="mwsrws" value="http://Hyland:8010/mws/mwsrepository" />
   ```

## Client-Konfiguration mwsrwsbasic

Die folgende WSDL-URL kann von einem Drittanbieter-Client wie z. B. einem JAVA-Client verwendet werden, um die Servicedefinition abzurufen.

```
http://localhost:8010/mwsbasic?wsdl
```

Diese Adresse setzt sich aus der host-base-Adresse gefolgt von ?wsdl zusammen.

**Beispiel**

Im folgenden Beispiel wird die WSDL URL des Server-Dienstes auf dem Computer mit dem Netzwerknamen `Hyland` am IP-Port `8010` angegeben:

```
http://Hyland:8010/mwsbasic?wsdl
```

## Konfigurieren des ComposerRepositoryWebServices (mwsrwsbasic) auf dem Server

Die spezifischen Funktionen dieses Dienstes werden durch den Dienst **Composer.MWS** bereitgestellt. Der Dienst wird als Windows-Dienst als Teil der Content Composer Standardinstallation installiert.

Um die individuellen Einstellungen für diesen Server-Endpunkt zu konfigurieren, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.MWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Zeile.

   ```
   <add baseAddress ="http://localhost:8010/mwsbasic" />
   ```

3. Prüfen Sie das Attribut `baseAddress` und ändern Sie falls notwendig den Wert.

   **Hinweis**: Als value muss dabei die host-base-Addresse des Services angegeben werden.

**Beispiel**

```
<add baseAddress ="http://Hyland:8010/mwsbasic" />
```

## RemoteControlService

### Informationen zu Remote Control

Die Content Composer-Standardinstallation fügt automatisch Einstellungen für den **Remote Control Client** für die folgenden Windows-Dienste hinzu.

- Composer.Core
- Composer.MWS
- Composer.OWS
- Composer.XWS

in den Dateien **Composer.Studio.exe.config** und **RemoteControl.config** hinzu und fügt ebenso die Default-Einstellungen für die Remote Control Dienste in den entsprechenden Dienst-Konfigurationsdateien (Composer.Core.exe.config, Composer.MWS.exe.config, etc.) hinzu. Diese müssen nur in Bezug auf den Hostnamen von Servern und Clients geändert werden.

Das folgende Beispiel veranschaulicht die erforderlichen Einstellungen für einen zusätzlichen Windows-Dienst:

**Dienstname**: ComposerProcess.SVC

**ServiceHost**: WindowsServiceHostModus.exe (umbenannte Kopie von Composer.WindowsServiceHost.exe)

### Client Konfiguration (Studio)

Die RemoteControl-Konfigurationsdatei wird über die ConfigSource **remotecontrol** in der Konfigurationsdatei **Composer.Studio.exe.config** angegeben. rework

```
<remotecontrol configSource="remotecontrol.config"/>
```

In der Konfigurationsdatei **RemoteControl.config** wird angegeben, welche RemoteControlServices auf welchem Server angesprochen werden.

In nachfolgendem Beispiel wird der Server `Hyland` und der Remote Control Service `RemoteControlServiceComposer` verwendet.

Dieser wird über den entsprechenden endpoint angegeben.

Der Windows-Dienst wird über den Namen des Dienstes (ModProcess_SVC) angegeben

```
<remotecontrol>
 <computers>
   <computer name="Hyland" display="Content Composer Server">
     <services>
       ...
       <service name="ModProcess_SVC"
```

```
                display="Composer Process Service"
                runtime="TRUE"
                endpoint="RemoteControlServiceComposer"/>
       ...
     </services>
   </computer>
  </computers>
</remotecontrol>
```

In der Konfigurationsdatei **Composer.Studio.exe.config** werden über das Element **<client>** die verwendeten endpoints angegeben.

Im folgenden Beispiel kommuniziert der Client mit dem Server-Dienst **RemoteControlService** auf Port **4712** auf dem Computer mit dem Netzwerknamen **Hyland**.

Der RemoteControlService wird dabei über seine endpoint-Adresse angegeben.

```
<system.serviceModel>
  ...
  <client>
    <endpoint name="RemoteControlServiceModus"
              address="http://Hyland:4712/RemoteControlService"
              binding="wsDualHttpBinding"
              bindingConfiguration="Binding1"

contract="ComposerSuite.Runtime.RemoteControlContract.IRuntimeControlServic
e"/>
   ...
  </client>
  ...
</system.serviceModel>
```

Desweiteren wird über das zugeordnete Binding (hier Binding1) die clientBaseAdresse angegeben, über die der Server mit dem Client kommuniziert.

Im folgenden Beispiel läuft der Client (RemoteControllClient im Studio) auf dem Computer mit dem Netzwerknamen **Composer_Client** am IP-Port **8000**:

```
</system.serviceModel>
 ...
 <bindings>
   <wsDualHttpBinding>
     <binding name="Binding1"
        clientBaseAddress="http://Composer_Client:8000/RemoteControlClient"
        ...
    </binding>
...
   </wsDualHttpBinding>
...
```

```
 </bindings>
...
</system.serviceModel>
```

### Server-Konfiguration

Das folgende Beispiel veranschaulicht die erforderlichen Einstellungen für einen zusätzlichen Windows-Dienst:

**Dienstname**: ModProcess_SVC

**ServiceHost**: WindowsServiceHostModus.exe (umbenannte Kopie von Composer.WindowsServiceHost.exe)

Dieser Dienst wird dabei dann durch den Windows-Service **WindowsServiceHostComposer** zur Verfügung gestellt.

Die Konfiguration des Server-Endpoints erfolgt daher in der Konfigurationsdatei **WindowsServiceHostComposer.config** mit Hilfe der folgenden XML-Elemente:

```
<remotecontrol address="http://Hyland:4712/RemoteControlService" />
<system.serviceModel>
 <services>
   ...
   <service name="ComposerSuite.Runtime.RemoteControlService">
       <!-- use base address provided by host -->
     <endpoint
       address="http://Hyland:4712/RemoteControlService"
       binding="wsDualHttpBinding"
       bindingConfiguration ="RemoteControlBinding"

contract="ComposerSuite.Runtime.RemoteControlContract.IRuntimeControlService"
     />
   </service>
   ...
 </services>
<system.serviceModel>
```

## RemoteDebugger

### Informationen zum Debug Service RemoteDebugger

Um das Debuggen von Skripten und Bedingungen im Content Composer Client zu ermöglichen, muss das Modul

```
<ModuleInfo AssemblyFile="ModusSuite.Studio.RemoteDebugger.dll" />
```

im Content Composer Client geladen werden.

Die vom Content Composer Client zu ladenden Module werden innerhalb der Konfigurationsdatei **clientprofile.xml** definiert.

## Client-Konfiguration (Client)

In der Konfigurationsdatei **Composer.Studio.exe.config** wird über das XML-Element **<client>** der Endpoint angegeben.

Ein Client benutzt den Endpoint-Namen nur zur Verbindung mit dem Debug-Dienst **ComposerSuite.Runtime.DebugService**.

Im folgenden Beispiel kommuniziert der Client mit dem Server-Dienst **DebugService** auf Port **8012** auf dem Computer mit dem Netzwerknamen **Hyland**. rework

Der **RemoteControlService** wird dabei über seine endpoint-Adresse angegeben.

```
<system.serviceModel>
 ...
 <client>
    <endpoint name="ComposerSuite.Runtime.DebugService"
       address=”http://Hyland:8012/RemoteDebugger”
       binding="wsDualHttpBinding"
       bindingConfiguration="DebugClientBinding"
       contract="ComposerSuite.Runtime.Types.IDebugService" />
   </client>
...
 </client>
 ...
</system.serviceModel>
```

Desweiteren wird über das **DebugClientBinding** die clientBaseAdresse angegeben, über welche der Server mit dem Client kommuniziert.

Im folgenden Beispiel läuft der Client (DebugClient im Content Composer Client) auf dem Computer mit dem Netzwerknamen **Hyland_Client** am IP-Port **4711**:

```
</system.serviceModel>
 ...
 <bindings>
   <wsDualHttpBinding>
     <binding name="DebugClientBinding"
              clientBaseAddress="http://Hyland_Client:4711/DebugClient"
              ...
    </binding>
...
   </wsDualHttpBinding>
...
 </bindings>
...
</system.serviceModel>
```

### Server-Konfiguration

Die spezifischen Funktionen dieses Dienstes werden durch den Dienst **Composer.MWS** bereitgestellt, der als Windows-Dienst als Teil der Content Composer-Standardinstallation installiert wird.

Die Konfiguration des Server- Endpoints erfolgt daher in der Konfigurationsdatei **Composer.MWS.exe.config**.

Der folgende Eintrag ist im `appSettings`-Element erforderlich:

```
<appSetting>
...
 <add key="disableDebugNamedPipe" value="1"/>
...
</appSetting>
```

Im folgenden Beispiel läuft der Debug-Service auf dem Computer mit dem Netzwerknamen **Hyland** am IP-Port **8012**:

```
<system.serviceModel>
 <services>
   ...
   <service name="ComposerSuite.Runtime.DebugService">
     <endpoint
         address="http://Hyland:8012/RemoteDebugger"
         binding="wsDualHttpBinding"
         bindingConfiguration ="DebugBinding"
         contract="ComposerSuite.Runtime.Types.IDebugService"/>
   </service>
  ...
 <services>
<system.serviceModel>
```

# Content Composer Zertifikate

## Informationen zu Zertifikaten

Das Content Composer Setup installiert die digitalen Zertifikate **composer.service** und **composer.server** in die Windows-Zertifikatspeicher **Local Computer** - **Personal** und Local Computer - Trusted People.

Content Composer verwendet diese Zertifikate, um den Kommunikationskanal zwischen den Content Composer Clients und den Content Composer Serverdiensten zu schützen.

Wenn die Sicherheitsrichtlinien Ihres Unternehmens die Verwendung von anwendungsspezifischen Zertifikaten nicht erlauben, befolgen Sie die Anweisungen in diesem Kapitel, um eigene Zertifikate zu erstellen und die Content Composer Zertifikate damit zu ersetzen.

## Austauschen der Zertifikate

Um die Content Composer Zertifikate auszutauschen, führen Sie die folgenden Aufgaben durch:

1. Schließen Sie auf allen Computern alle Content Composer Anwendungen.
2. Stoppen Sie auf jedem Content Composer Server den **Composer.Core** Dienst. Dadurch werden auch alle anderen Content Composer Dienste auf diesem Computer gestoppt.
3. Entfernen sie das Root-Zertifikat. Weitere Informationen finden Sie im Kapitel Entfernen des Root-Zertifikats composer.CA.
4. Erstellen Sie die neuen Zertifikate. Weitere Informationen finden Sie im Kapitel Erstellen der Zertifikate.
5. Installieren Sie die neuen Zertifikate. Weitere Informationen finden Sie im Kapitel Installieren der Zertifikate.
6. Passen Sie die Server-Konfiguration an. Weitere Informationen finden Sie im Kapitel Anpassen der Server-Konfiguration.
7. Passen Sie die Studio-Konfiguration an. Weitere Informationen finden Sie im Kapitel Anpassen der Studio-Konfiguration.
8. Passen Sie die Client-Konfiguration an. Weitere Informationen finden Sie im Kapitel Anpassen der Client-Konfiguration.
9. Prüfen Sie die Funktionalität. Weitere Informationen finden Sie im Kapitel Prüfen der Funktionalität.
10. Falls keine Problem aufgetreten sind, löschen Sie auf jedem Computer mit einer Content Composer Server-, Studio- oder Windows Client-Installation die Zertifikate
    - **composer.server** und
    - **composer.service**

    aus den Zertifikatspeichern
    - **Local Computer - Personal - Zertifikate** und
    - **Local Computer - Trusted People - Zertifikate**
11. Starten Sie jede Instanz von Content Composer Studio und Windows Client neu.

### Entfernen des Root-Zertifikats *composer.CA*

Das Content Composer-Setup für Foundation EP1 und vorhergehende Versionen installiert das Root-Zertifikat **composer.CA**.

Ab Foundation EP2 wird dieses Zertifikat nicht mehr benötigt. Führen Sie folgenden Schritt aus:

- Löschen Sie auf jedem Computer mit einer Content Composer Server-, Studio- oder Windows Client-Installation das Zertifikat **composer.CA** aus dem Zertifikatspeicher **Local Computer - Trusted Root Certification Authorities - Zertifikate**.

### Erstellen der Zertifikate

Um die Zertifikate zu erstellen, führen Sie auf dem Content Composer Server die folgenden Schritte aus:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%admin\certs** die Datei CreateNewComposerCerts.ps1 mit einem Texteditor, der UTF-8 unterstützt.

2. Suchen Sie die folgenden Zeile:

   ```
   $certPassword = ConvertTo-SecureString -String "pwd" -Force -
   AsPlainText
   ```

3. Ersetzen Sie `pwd` mit dem Passwort, das Sie für Ihre PFX-Dateien verwenden wollen.
4. Speichern und schließen Sie die Datei.
5. Öffnen Sie ein CMD-Fenster mit Administratorrechten und wechseln Sie in das Verzeichnis **%Composerdir%admin\certs**.
6. Führen Sie folgenden Befehl aus:

   ```
   powershell -file CreateNewComposerCerts.ps1
   ```

   **Ergebnis**: Die folgenden Dateien sind in Ihrem Windows-Dokumentenordner verfügbar:

   - **composernew_CA.cer**: Dieses Root-Zertifikat wird nur zur Erstellung der anderen Zertifikate benötigt.
   - **composer_servernew.pfx**: Ersetzt das Zertifikat **composer.server**.
   - **composer_servicenew.pfx**: Ersetzt das Zertifikat **composer.service**.

   **Hinweis**: Das Skript gibt nach erfolgreicher Ausführung den Speicherort der Dateien aus.

### Installieren der Zertifikate

Um die Zertifikate zu installieren, führen Sie die folgenden Schritte durch.

1. Öffnen Sie auf dem Content Composer Server aus dem Verzeichnis **%Composerdir%admin\certs** die Datei InstallNewComposerCerts.ps1 mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Zeile:

   ```
   $certPassword = ConvertTo-SecureString -String "pwd" -Force -
   AsPlainText
   ```

3. Ersetzen Sie `pwd` mit dem selben Passwort, das Sie beim Erstellen der Zertifikate verwendet haben.
4. Speichern und schließen Sie die Datei.
5. Führen Sie die folgenden Teilschritte auf jedem Computer mit einer Content Composer Server-, Studio- oder Windows Client-Installation durch:
   1. Kopieren Sie die folgenden Dateien in ein Verzeichnis unterhalb von **%Composerdir%**:
      - InstallNewComposerCerts.ps1
      - composer_servernew.pfx
      - composer_servicenew.pfx
   2. Öffnen Sie als Administrator ein CMD-Fenster und wechseln Sie in das Verzeichnis, in das Sie die Dateien im vorherigen Schritt kopiert haben.
   3. Führen Sie folgenden Befehl aus:

      ```
      powershell –file InstallNewComposerCerts.ps1
      ```

**Ergebnis**: Die beiden Zertifikate sind in den folgenden Windows-Zertifikatsspeichern verfügbar:

- Local Computer - Personal - Zertifikate
- Local Computer - Trusted People - Zertifikate

## Anpassen der Server-Konfiguration

Um den Content Composer Server für die Verwendung der neuen Zertifikate anzupassen, führen Sie die folgenden Schritte durch:

1. Kopieren Sie auf dem Content Composer Server alle CONFIG-Dateien aus dem Verzeichnis **%Composerdir%** in ein Sicherungsverzeichnis.
2. Führen Sie die folgenden Teilschritte für die folgenden Dateien durch.
   - Composer.Core.exe.config
   - Composer.MWS.exe.config
   - Composer.OWS.exe.config
   - Composer.XWS.exe.config
   - Composer.WCF.Console.exe.config

   1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die CONFIG-Datei mit einem Texteditor, der UTF-8 unterstützt.
   2. Suchen Sie die folgenden Zeile:

      ```
      <add storeLocation="LocalMachine" storeName="TrustedPeople"
      x509FindType="FindBySubjectDistinguishedName"
      findValue="CN=composer.server" />
      ```

   3. Ändern Sie `composer.server` in `composer.servernew`.

      ```
      <add storeLocation="LocalMachine" storeName="TrustedPeople"
      x509FindType="FindBySubjectDistinguishedName"
      findValue="CN=composer.servernew" />
      ```

   4. Suchen Sie die folgenden Zeile:

      ```
      <serviceCertificate storeLocation="LocalMachine"
      storeName="My" x509FindType="FindBySubjectDistinguishedName"
      findValue="CN=composer.service" />
      ```

   5. Ändenr Sie `composer.service` in `composer.servicenew`.

      ```
      <add storeLocation="LocalMachine" storeName="TrustedPeople"
      x509FindType="FindBySubjectDistinguishedName"
      findValue="CN=composer.servicenew" />
      ```

   6. Speichern und schließen Sie die Datei.
3. Führen Sie die folgenden Teilschritte für die folgenden Dateien durch.
   - Composer.Core.exe.config
   - Composer.MWS.exe.config

- Composer.OWS.exe.config
- Composer.XWS.exe.config
- Composer.WCF.Console.exe.config
- Composer.WindowsServiceHost.exe.config
- Alle benutzerdefinierten Content Composer Service-Konfigurationsdateien

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die CONFIG-Datei mit einem Texteditor, der UTF-8 unterstützt.
2. Fügen Sie innerhalb des `<appsettings>` Elements die folgenden Zeilen ein:

```
<add key="certStsSubjectName" value="composer.servernew" />
<add key="certServiceSubjectName" value="composer.servicenew"
/>
```

3. Speichern und schließen Sie die Datei.

### Anpassen der Studio-Konfiguration

Um das Content Composer Studio für die Verwendung der neuen Zertifikate anzupassen, führen Sie die folgenden Schritte durch:

1. Führen Sie die folgenden Schritte auf einem Computer aus, auf dem das Content Composer Studio installiert ist.
2. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Studio.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
3. Fügen Sie innerhalb des `<appsettings>` Elements die folgenden Zeilen ein:

```
<add key="save_login_token" value="True" />
<add key="certStsSubjectName" value="composer.servernew" />
<add key="certServiceSubjectName" value="composer.servicenew" />
```

4. Speichern und schließen Sie die Datei.
5. Starten Sie Content Composer Studio und beenden Sie es anschließend wieder.
6. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Studio.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
7. Suchen Sie die folgenden Zeile und löschen Sie diese:

```
<add key="save_login_token" value="True" />
```

8. Speichern und schließen Sie die Datei.
9. Kopieren Sie aus dem Verzeichnis **%Composerdir%** die Datei **securitytoken.xml** in das **%Composerdir%**-Verzeichnis jedes Computers, auf dem Content Composer Server, Studio oder der Windows Client installiert ist.
10. Führen Sie die folgenden Teilschritte auf jedem Computer, auf dem das Content Composer Studio installiert ist, durch:
    1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Studio.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
    2. Fügen Sie innerhalb des `<appsettings>` Elements die folgenden Zeilen ein:

```
<add key="certStsSubjectName" value="composer.servernew" />
<add key="certServiceSubjectName" value="composer.servicenew"
/>
```

3. Suchen Sie die folgende Zeile und falls vorhanden, löschen Sie diese:

```
<add key="save_login_token" value="True" />
```

4. Speichern und schließen Sie die Datei.

### Anpassen der Client-Konfiguration

Um die Content Composer Windows Clients für die Verwendung der neuen Zertifikate anzupassen, führen Sie die folgenden Schritte auf jedem Computer aus, auf dem der Content Composer Client installiert ist:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Client.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Fügen Sie innerhalb des `<appsettings>` Elements die folgenden Zeilen ein:

```
<add key="certStsSubjectName" value="composer.servernew" />
<add key="certServiceSubjectName" value="composer.servicenew" />
```

3. Speichern und schließen Sie die Datei.

## Prüfen der Funktionalität

Um zu überprüfen, ob die neuen Zertifikate korrekt funktionieren, führen Sie die folgenden Schritte durch:

1. Starten Sie auf jedem Content Composer Server die folgenden Windows-Dienste neu:
   - Composer.Core
   - Composer.MWS
   - Composer.OWS
   - Composer.XWS (falls nötig)
2. Um zu überprüfen, ob das Content Composer Studio und die Windows Clients mit den Content Composer-Diensten kommunizieren können, führen Sie die folgenden Teilschritte durch:
   1. Starten Sie auf jedem Computer, auf dem das Content Composer Studio installiert ist, das Studio.
   2. Öffnen Sie ein beliebiges Objekt aus dem **Navigator** und überprüfen Sie das Ausgabefenster auf Fehler.
   3. Starten Sie auf jedem Computer, auf dem der Content Composer Client installiert ist, den Client.
   4. Öffnen Sie die **Prozessliste** und überprüfen Sie das Ausgabefenster auf Fehler.

# Konfiguration des Loggings

## Informationen zum Logging

Protokolldateien helfen Ihnen bei der Analyse der Ursache von z. B. unerwartetem Verhalten oder plötzlicher Anwendungsbeendigung.

Content Composer unterstützt .NET Framework Logging, Hyland Logging, Audit Logging und Splunk Logging.

## Informationen zum .NET Framework Logging

Dieser Logging-Typ ist in einer Standardinstallation aktiviert und verwendet die .NET-Logging-Implementierung, die im .NET-Namespace **System.Diagnostics** enthalten ist.

### Aktivieren des .NET Framework-Loggings

So aktivieren Sie das .NET Framework Logging:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die entsprechende Konfigurationsdatei, z. B. **Composer.Studio.exe.config**, mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie nach dem `<appSettings>` Element.
3. Fügen Sie innerhalb des `<appSettings>`-Elements die folgende Zeile hinzu oder ändern Sie sie, indem Sie das Attribut `value` auf `False` setzen.

   ```
   <add key="EnableHylandLogging" value="False" />
   ```

4. Speichern und schließen Sie die Datei.

### Standard-Speicherort der Logdatei

- Um sicherzustellen, dass der Benutzer über ausreichende Rechte zum Erstellen und Schreiben in die Logdatei verfügt, befinden sich die Logdateien standardmäßig im Verzeichnis **%userprofile%\ContentComposer**.
- Wenn die Dienste einen Windows-Dienstaccount verwenden, befinden sich die Logdateien des Dienstes standardmäßig im Verzeichnis **%Composerdir%**.

Der Name der Logdatei wird um das Datum ergänzt.

**Beispiel**: *Composer.Studio_Log_20201116.txt*

### Ändern des Speicherorts der Log-Dateien

Um den Speicherort der Log-Dateien zu ändern, führen Sie den folgenden Schritt aus:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die entsprechende CONFIG-Datei mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie innerhalb des `<sharedListeners>` Elements, nach der Zeile, die wie folgt beginnt:

   ```
   <add name="tracefile_listener" type=
   ```

3. Setzen Sie den Wert des `initializeData` Attributs auf den Dateiname der Log-Datei. Bei

Bedarf können Sie auch einen Pfad angeben.

Weitere Informationen finden Sie unter Standard-Speicherort der Logdatei.

**Beispiel**

```
<add name="tracefile_listener"
type="ModusSuite.Common.SystemFramework.DailyTraceListener,
ModusSuite.Common.SystemFramework" delimiter=";"
initializeData="C:\CoCoLogFiles\Composer.Studio_Log.txt"
traceOutputOptions="DateTime" />
```

4. Speichern und schließen Sie die Datei.

## Ändern des Log-Levels

Um das Log-Level zu ändern, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die entsprechende CONFIG-Datei mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie nach dem `<switches>` Element.
3. Setzen Sie innerhalb des `<switches>` Elements das `value` Attribut des Switches, den Sie ändern möchten, auf das gewünschte Log Level.

| Log Level | Beschreibung |
|---|---|
| Off | Das Logging ist deaktiviert. |
| Error | Dieses Level protokolliert nur Fehler. |
| Warning | Dieses Level protokolliert Warnungen und Fehler.<br>Warnungen sind keine Fehler, weisen jedoch auf fehlernahe Zustände, wie z.B. fehlende Daten hin. Z. B. sind fehlende Daten nicht notwendigerweise ein Fehler, sollten jedoch protokolliert werden. |
| Information | Dieses Level protokolliert zusätzlich zu den Fehlermeldungen und Warnungen auch informative Meldungen. |
| Verbose | Dieses Level schreibt detaillierte Informationen über alle Prozessaktivitäten in die Logdatei. |

**Hinweis**: Beim Wert wird zwischen Groß- und Kleinschreibung unterschieden.

**Beispiel**

```
add name="studio_switch" value="Verbose" />
```

4. Speichern und schließen Sie die Datei.
5. Starten Sie den entsprechenden Dienst neu.

## Informationen zum Hyland Logging

Das Hyland Logging verwendet die .NET Core Logging-Implementierung, die im Namespace **Microsoft.Extensions.Logging** enthalten ist. Dieses basiert auf dem Open-Source-Logging-Framework **Serilog**.

### Aktivieren des Hyland-Loggings

So aktivieren Sie das Hyland Logging:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die entsprechende Konfigurationsdatei, z. B. **Composer.Studio.exe.config**, mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie nach dem `<appSettings>` Element.
3. Fügen Sie innerhalb des `<appSettings>`-Elements die folgende Zeile hinzu oder ändern Sie sie, indem Sie das Attribut `value` auf `True` setzen.

   ```
   <add key="EnableHylandLogging" value="True" />
   ```

4. Speichern und schließen Sie die Datei.

### Routen-Konfiguration

Eine Konfigurationsdatei einer Content Composer-Anwendung enthält ein oder mehrere `<Route>`-XML-Elemente.

Ein `<Route>`-Element enthält die folgenden Konfigurations-Keys:

| Key-Name | Beschreibung |
|---|---|
| minimum-level | Legt das minimale `LogLevel` fest, das in diese Route geschrieben wird.<br>**Mögliche Werte**<br>• Error<br>• Warning<br>• Information<br>• Debug |
| File | Pfad zur Logdatei einschließlich des Dateinamens. Der Pfad unterstützt Umgebungsvariablen.<br>**Beispiel**<br>`%composerdir%\Composer_Core_log.txt` |

| Key-Name | Beschreibung |
|---|---|
| FileRollInterval | Ermöglicht das Rollieren von Dateien im angegebenen Zeitintervall.<br>**Mögliche Werte**<br>• Day<br>• Hour<br>• Minute<br>• Month<br>• Year<br>• Infinite |
| FileRollOnSize | Ermöglicht das Rollieren von Dateien, wenn das `FileByteLimit` erreicht ist.<br>Wenn der Key vorhanden ist, wird die Datei abhängig von der Größe rolliert, außer der Wert wird ausdrücklich auf `False` gesetzt.<br>**Mögliche Werte**<br>• True<br>• False |
| FileByteLimit | Legt die maximale Dateigröße für eine einzelne Logdatei fest.<br>Wenn `FileRollOnSize` nicht enthalten oder auf `False` gesetzt ist, werden zusätzliche Nachrichten bis zum nächsten Rollierungs-Intervall verworfen, falls zutreffend.<br>Das Standard-Bytelimit für Dateien beträgt 1 GB. Um unbegrenzte Dateigrößen zuzulassen, setzen Sie den Wert auf eine leere Zeichenfolge. |
| OutputFormat | Legt das Ausgabeformat der Zeilen in der Logdatei fest<br>**Mögliche Werte**<br>• **Minimal**: Protokolliert nur Zeit, Log-Level, Exception und Meldung und schreibt keine weiteren Felder in die Nachricht.<br>• **Text**: Protokolliert alle Felder der Nachricht als eine Liste von Schlüssel-Wert-Paaren. Diese ist besser lesbar als die Option `Json` und enthält mehr Informationen als die Option `Minimal`.<br>• **Json**: Schreibt die Meldung in kompaktem Json-Format in eine einzige Zeile. Diese enthält so viele Informationen wie möglich, genau wie bei `Text`, aber in einer weniger gut lesbaren Form. Dieses Format ist ideal für die weitere Verarbeitung.<br>Der Defaultwert ist `Json`. |

| Key-Name | Beschreibung |
|---|---|
| exclude-profiles | Enthält die Profile die nicht in diese Route geschrieben werden als kommagetrennte Liste. Alle nicht in der Liste enthaltenen Profile werden protokolliert.<br>Die Default-Einstellung ist ein Leerstring.<br>**Hinweis**: Der Key `include-profiles` hat Vorrang vor dieser Einstellung. |
| include-profiles | Definiert die Profile die in diese Route geschrieben werden als kommagetrennte Liste. Alle nicht in der Liste enthaltenen Profile werden nicht protokolliert.<br>Um alle Profile einzuschließen, setzen Sie den Wert auf eine leere Zeichenfolge. |
| AllowUnknownKeys | Ist dieser Schlüssel vorhanden, werden unbekannte Konfigurations-Keys an die **Serilog**-Library übergeben. |

## Profile

Die Content Composer-Komponenten verwenden unterschiedliche Profilnamen, wenn sie eine Meldung in die Protokolldatei schreiben.

Um Profile ein- oder auszuschließen, fügen Sie diese Profilnamen den Schlüsseln `include-profiles` und `exclude-profiles` innerhalb einer `<Route>`-Konfiguration hinzu.

Die folgende Tabelle listet die verfügbaren Profilnamen auf:

| Profilname | Beschreibung |
|---|---|
| mws | MWS Dienst |
| rws | Repository Service |
| modusstudio | Studio GUI |
| mwsclient | Content Composer Windows Client |
| monalisa | Monalisa Scripting |
| runtime_process | Content Composer Process Engine |
| runtime_components | Content Composer Process Komponenten |
| textsystem | Dokumentenerstellung |

| Profilname | Beschreibung |
|---|---|
| xdata | Xdata |
| Odin | Odin |
| mur | Benutzer-Authentifizierung und -Autorisierung |
| dpws | Data Provider |

### Verfügbare Route-Konfigurationen

Die Standardkonfigurationsdateien von Content Composer bieten vordefinierte Routen für bestimmte Logging-Zwecke.

Die folgende Tabelle beschreibt den Zweck der einzelnen Routen:

| Route-Name | Beschreibung |
|---|---|
| CoCo_Standard | Die Standard-Route enthält alle Profile, die nur Fehler protokollieren. Jede Anwendungs-Konfigurationsdatei enthält dieses Basisprofil. |
| CoCo_Runtime_ Process | Diese Route konfiguriert das minimale Log-Level `Information` für das Profil `runtime_process`. |
| CoCo_ OutputWindow | Diese Route konfiguriert das minimale Log-Level `Information` für alle Profile, die im Ausgabefenster im Content Composer Studio oder Content Composer Windows Client protokolliert werden. |
| CoCo_Monalisa | Diese Route konfiguriert das minimale Log-Level `Information` für das Profil `monalisa`. |
| CoCo_ Textsystem | Diese Route konfiguriert das minimale Log-Level `Information` für das Profil `textsystem`. |

## Informationen zum Audit Logging

Das Audit Logging ist für bestimmte Benutzeraktionen in Content Composer Studio, Content Composer Web Client und Odin-Benutzeraktionen in Content Composer Studio verfügbar.

### Aktivieren des Audit Loggings

So aktivieren Sie das Audit Logging:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die entsprechende CONFIG-Datei mit einem Texteditor, der UTF-8 unterstützt.

- Um das Audit Logging für Studio-Benutzeraktionen zu aktivieren, öffnen Sie **Composer.Studio.exe.config**.
- Um das Audit Logging für Web Client Benutzeraktionen zu aktivieren, öffnen Sie **Composer.MWS.exe.config**.
- Um das Audit Logging für Odin-Benutzeraktionen im Studio zu aktivieren, öffnen Sie **Composer.OWS.exe.config**.

2. Suchen Sie nach dem `<AuditRoutes>` Element.
3. Fügen Sie innerhalb des Elements `<AuditRoutes>` die folgende Zeilen ein.

```
<Route name="CoCo_Audit">
   <add key="minimum-level" value="Information"/>
   <add key="File" value="%composerdir%\Composer.AUDIT.txt" />
   <add key="FileRollInterval" value="Day"/>
   <add key="FileRollOnSize" value="true"/>
   <add key="FileByteLimit" value="1000000" />
   <add key="FileCountLimit" value="31" />
   <add key="OutputFormat" value="Json" />
   <add key="AllowUnknownKeys" value="" />
</Route>
```

4. Optional. Ändern Sie die Werte bei Bedarf.

   Weitere Informationen finden Sie unter Audit Logging Konfiguration.
5. Speichern und schließen Sie die Datei.
6. Starten Sie die folgenden Dienste neu:
   - Composer.Core
   - Composer.MWS
   - Composer.OWS

## Audit Logging Konfiguration

Ein `<AuditRoutes>`-Element enthält die folgenden Konfigurations-Keys:

| Key-Name | Beschreibung |
|---|---|
| minimum-level | Legt das minimale `LogLevel` fest, das in diese Route geschrieben wird.<br>**Mögliche Werte**<br>• Error<br>• Warning<br>• Information<br>• Debug |
| File | Pfad zur Logdatei einschließlich des Dateinamens. Der Pfad unterstützt Umgebungsvariablen. |

| Key-Name | Beschreibung |
|---|---|
| | **Beispiel**<br>`%composerdir%\Composer.AUDIT.txt`<br>**Hinweis**: Der Name der Datei wird um das Datum ergänzt. |
| FileRollInterval | Ermöglicht das Rollieren von Dateien im angegebenen Zeitintervall.<br>**Mögliche Werte**<br>• Day<br>• Hour<br>• Minute<br>• Month<br>• Year<br>• Infinite |
| FileRollOnSize | Ermöglicht das Rollieren von Dateien, wenn das `FileByteLimit` erreicht ist.<br>Wenn der Key vorhanden ist, wird die Datei abhängig von der Größe rolliert, außer der Wert wird ausdrücklich auf `False` gesetzt.<br>**Mögliche Werte**<br>• True<br>• False |
| FileByteLimit | Legt die maximale Dateigröße für eine einzelne Logdatei fest.<br>Wenn `FileRollOnSize` nicht enthalten oder auf `False` gesetzt ist, werden zusätzliche Nachrichten bis zum nächsten Rollierungs-Intervall verworfen, falls zutreffend.<br>Das Standard-Bytelimit für Dateien beträgt 1 GB. Um unbegrenzte Dateigrößen zuzulassen, setzen Sie den Wert auf eine leere Zeichenfolge. |
| FileCountLimit | Legt die maximale Anzahl der Dateien fest, die bei der Verwendung von rollierenden Dateien aufbewahrt werden. Um eine unbegrenzte Anzahl von Dateien zuzulassen, verwenden Sie eine leere Zeichenfolge. |
| OutputFormat | Legt das Ausgabeformat der Zeilen in der Logdatei fest<br>**Mögliche Werte**<br>• **Minimal**: Protokolliert nur Zeit, Log-Level, Exception und Meldung und schreibt keine weiteren Felder in die Nachricht.<br>• **Text**: Protokolliert alle Felder der Nachricht als eine Liste von Schlüssel- |

| Key-Name | Beschreibung |
|---|---|
| | Wert-Paaren. Diese ist besser lesbar als die Option `Json` und enthält mehr Informationen als die Option `Minimal`.<br>• **Json**: Schreibt die Meldung in kompaktem Json-Format in eine einzige Zeile. Diese enthält so viele Informationen wie möglich, genau wie bei `Text`, aber in einer weniger gut lesbaren Form. Dieses Format ist ideal für die weitere Verarbeitung.<br>Der Defaultwert ist `Json`. |
| AllowUnknownKeys | Ist dieser Schlüssel vorhanden, werden unbekannte Konfigurations-Keys an die **Serilog**-Library übergeben. |

## Protokollierte Benutzeraktionen im Studio

Die folgenden Benutzeraktionen in Content Composer Studio werden protokolliert, wenn die Audit-Logging aktiviert ist.

**Allgemein**

| Benutzeraktion | Action Nummer | SubAction Nummer |
|---|---|---|
| Geändertes Objekt speichern | 20 | 2 |
| Neue Version erstellen | 20 | 10 |

**Navigator Kontextmenu**

| Benutzeraktion | Action Nummer | SubAction Nummer |
|---|---|---|
| Neu | 20 | 1 |
| Löschen (Objekt) | 20 | 6 |
| Kopieren \| Einfügen | 20 | 3 |
| Kopieren mit untergeordneten Objekten \| Einfügen | 20 | 4 |
| Umbenennen | 20 | 5 |
| Objekthistorie anzeigen \| Eine alte Version aus der Liste auswählen \| Objekt öffnen \| Version reaktivieren | 20 | 11 |

**Navigator Toolbar**

| Benutzeraktion | Action Nummer | SubAction Nummer |
|---|---|---|
| Gelöschte Objekte \| Dauerhaftes Löschen von Objekten | 20 | 6 |
| Gelöschte Objekte \| Objekte wiederherstellen | 20 | 9 |

**Systemverwaltung**

| Benutzeraktion | Action Nummer | SubAction Nummer |
|---|---|---|
| Neues System | 21 | 1 |
| Neuer DB-Alias | 21 | 1 |
| Ändern und speichern der Eigenschaften eines Systems im Objektinspektor | 21 | 2 |
| Löschen eines Systems | 21 | 6 |
| Wiederherstellen eines Systems | 21 | 9 |
| Ändern der Einstellungen eines Datenbank-Alias | 21 | 2 |
| Löschen eines Datenbank-Alias | 21 | 6 |

**Transferjobs**

| Benutzeraktion | Action Nummer | SubAction Nummer |
|---|---|---|
| Einen geplanten Transfer-Job erstellen | 22 | Start: 19<br>Ende: 20 |
| Ändern eines geplanten Transfer-Jobs | 22 | 17 |
| Löschen eines geplanten Transfer-Jobs | 22 | 18 |
| Ausführen eines Transfer-Jobs | 22 | Start: 13<br>Ende: 14 |

| Benutzeraktion | Action Nummer | SubAction Nummer |
|---|---|---|
| Einen Transfer widerrufen | 22 | Start: 15<br>Ende: 16 |
| Repository Import | 22 | Start: 12<br>Ende: 14 |

## Protokollierte Benutzer-Aktionen im Web Client

Die folgenden Benutzeraktionen im Content Composer Web Client werden protokolliert, wenn das Audit-Logging aktiviert ist:

| Benutzeraktion | Action Nummer | SubAction Nummer |
|---|---|---|
| Anmelden | 30 | 1 |
| Abmelden | 30 | 2 |
| Erstellen einer neuen Kommunikation | 31 | 3 |
| Löschen einer Kommunikation | 31 | 4 |
| Weiterleiten einer Kommunikation | 31 | 5 |
| Bearbeiten und speichern einer DOCX-Datei | 31 | 6 |
| Ändern des Status einer Kommunikation (Öffnen / Schließen / Starten) | 31 | Öffnen: 7<br>Schließen: 8<br>Start: 9 |

## Protokollierte Odin-Benutzer-Aktionen

Die folgenden Benutzeraktionen in den Odin-Ansichten werden protokolliert, wenn das Audit-Logging aktiviert ist:

**Odin Ansicht Prozesse**

| Benutzeraktion | Action Nummer | SubAction Nummer |
|---|---|---|
| Prozess sperren | 1 | 1 |
| Prozess Lock aufheben | 1 | 2 |
| Status zurücksetzen auf | 1 | 10 |
| Status auf Löschen setzen | 1 | 8 |
| Stapel aus offenen Jobs erstellen | 2 | 5 |

**Odin Ansicht Stapel**

| Benutzeraktion | Action Nummer | SubAction Nummer |
|---|---|---|
| Stapel zurücksetzen | 2 | 6 |
| Stapel freigeben | 2 | 7 |
| Stapel löschen | 2 | 8 |
| Stapel sperren | 2 | 1 |
| Stapel Lock aufheben | 2 | 2 |
| Stapel für Streaming zurücksetzen | 2 | 9 |
| Stapel zurücksetzen auf | 2 | 10 |

**Odin Ansicht Stapel - Stapel bearbeiten**

| Benutzeraktion | Action Nummer | SubAction Nummer |
|---|---|---|
| Drucker auswählen für | 3 | 11 |
| Ändern der **Priorität** \| Speichern | 2 | 12 |
| Ändern der **Haltezeit** \| Speichern | 2 | 13 |

**Odin Ansicht Stapel - Kuverts**

| Benutzeraktion | Action Nummer | SubAction Nummer |
|---|---|---|
| Kuvert aus dem Stapel entfernen | 2 | 14 |

**Odin Ansicht Stapel - Jobs**

| Benutzeraktion | Action Nummer | SubAction Nummer |
|---|---|---|
| Drucker auswählen | 3 | 11 |
| Job zurücksetzen auf | 3 | 15 |
| Job entfernen | 2 | 17 |

**Odin Ansicht Stapel - Offene Jobs**

| Benutzeraktion | Action Nummer | SubAction Nummer |
|---|---|---|
| Stapel erstellen | 2 | 5 |

**Odin Ansicht Stapel - Offene Kuverts**

| Benutzeraktion | Action Nummer | SubAction Nummer |
|---|---|---|
| Freigeben | 4 | 8 |

## Deaktivieren des Audit Loggings

So deaktivieren Sie das Audit Logging:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die entsprechende CONFIG-Datei mit einem Texteditor, der UTF-8 unterstützt.
   - Um das Audit Logging für Studio-Benutzeraktionen zu deaktivieren, öffnen Sie **Composer.Studio.exe.config**.
   - Um das Audit Logging für Web Client Benutzeraktionen zu deaktivieren, öffnen Sie **Composer.MWS.exe.config**.
   - Um das Audit Logging für Odin-Benutzeraktionen im Studio zu deaktivieren, öffnen Sie **Composer.OWS.exe.config**.
2. Suchen Sie nach dem `<AuditRoutes>` Element.
3. Löschen Sie das `<Route>` Element innerhalb des `<AuditRoutes>` Elements.
4. Speichern und schließen Sie die Datei.
5. Starten Sie die folgenden Dienste neu:

- Composer.Core
- Composer.MWS
- Composer.OWS

## Informationen zum Splunk Logging

In diesem Abschnitt wird beschrieben, wie Sie Content Composer so konfigurieren, dass Protokollinformationen an Splunk gesendet werden.

### Konfigurieren von Content Composer für Splunk

So konfigurieren Sie Content Composer, um Log-Informationen an Splunk zu senden:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die entsprechende Konfigurationsdatei, z. B. **Composer.Studio.exe.config**, mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie nach dem `<AuditRoutes>` Element.
3. Fügen Sie innerhalb des Elements `<AuditRoutes>` die folgende Zeilen ein.

```
<Route name="Logging_Community_Splunk_coco" >
   <add key="Splunk"
value="http://localhost:8088/services/collector"/>
   <add key="SplunkToken" value="SplunkTokenNumber"/>
   <add key="minimum-level" value="Debug"/>
   <add key="include-profiles" value="mws" />
</Route>
```

4. Ersetzten Sie `localhost` mit der Adresse des **Splunk**-Servers.
5. Ersetzen Sie das value-Attribut von `SplunkToken` durch das Splunk-Token.
6. Optional. Ändern Sie die Werte für `minimum-level` und `include-profiles` gemäß Ihren Anforderungen.

   Weitere Informationen finden Sie unter Routen-Konfiguration.
7. Optional. Fügen Sie bei Bedarf weitere Keys hinzu.

   Weitere Informationen finden Sie unter Splunk Route Konfigurations-Keys.
8. Speichern und schließen Sie die Datei.
9. Starten Sie die entsprechenden Anwendung neu.
10. Starten Sie die folgenden Dienste neu:
    - Composer.Core
    - Composer.MWS
    - Composer.OWS

### Splunk Route Konfigurations-Keys

Die folgende Tabelle listet die für die Splunk-Routenkonfiguration verfügbaren Schlüssel auf:

| Key | Beschreibung |
|---|---|
| Splunk | Adresse des **Splunk**-Servers.<br>**Beispiel**<br>`http://[splunk-server]:8088/services/collector` |
| SplunkToken | Token, das vom Http Event Collector (HEC) verwendet wird, um Daten und Anwendungsereignisse an den Splunk-Server zu senden. |
| SplunkIndex (optional) | Index-Name des Tokens. |
| SplunkSource (optional) | Name der Datei, des Streams oder einer anderen Eingabe, von der ein bestimmtes Ereignis ausgeht. |
| SplunkHost (optional) | Name des Splunk-Hosts. |
| SplunkSourceType (optional) | Legt fest, wie Splunk den eingehenden Datenstrom je nach Art der Daten in einzelne Ereignisse verarbeitet. |

# Konfiguration der Benutzerverwaltung

## Die Benutzerverwaltung

Dieses Kapitel beschreibt die für die Authentifizierung und Autorisierung der Benutzer verfügbaren Optionen.

Die Authentifizierung prüft den Benutzernamen sowie das Passwort, während die Autorisierung die Benutzergruppen in der Benutzerverwaltung ermittelt.

Diese Gruppen werden den Content Composer Rollen zugewiesen, um die Benutzerrechte in den Content Composer Anwendungen zu ermitteln.

Content Composer verfügt über keine eigene Benutzerverwaltung. Stattdessen verwendet Content Composer eine der folgenden Benutzerverwaltungen:

- Windows Domain oder Workgroup
- LDAP

## Konfigurieren der Windows-Authentifizierung mit einer Workgroup oder einem Domain-Controller

Diese Konfigurationsvariante benutzt die Windows-Benutzerverwaltung zur Authentifizierung und Autorisierung der Benutzer.

Verwenden Sie diese Option, wenn kein LDAP-Server verfügbar ist.

Um die Windows-Authentifizierung unter Verwendung einer Workgroup oder eines Domain-Controllers zu konfigurieren, führen Sie die folgenden Schritte durch.

1. Aktivieren der Windows-Authentifizierung. Weitere Information finden Sie unter Aktivieren der Windows-Authentifizierung.
2. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **UserRepository.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
3. Setzen Sie den Wert des Attributs **profileReadOption** auf **None**.
4. Speichern und schließen Sie die Datei.

**Result** Für Benutzer und Rollen sind keine Profilattribute verfügbar. Die konfigurierten Attribute sind nur mit den angegebenen Defaultwerten verfügbar. Das System ermittelt zu Zuweisung von Benutzern zu Rollen über Windows.

## Konfigurieren der Windows-Authentifizierung durch einen Domaincontroller mit LDAP

Diese Konfigurationsvariante benutzt die Windows-Benutzerverwaltung (Domain oder Workgroup) zur Authentifizierung der Benutzer.

Zur Autorisierung konfigurieren Sie entweder die Windows-Benutzerverwaltung oder LDAP.

Falls nötig, liest das System die Benutzer- und Rollenattribute vom LDAP-Server.

Führen Sie einen der folgenden Teilschritte durch.

- Wenn das System die Windows-Benutzerverwaltung zur Authentifizierung und Autorisierung verwendet, führen Sie die folgenden Teilschritte aus, um die Zuordnung von Benutzern zu Rollen über Windows zu ermitteln.
  - Aktivieren der Windows-Authentifizierung. Weitere Information finden Sie unter Aktivieren der Windows-Authentifizierung.
  - Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **UserRepository.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
  - Setzen Sie den Wert des Attributs **profileReadOption** auf **All**, **Role**, **None** oder **User**. **Beispiel**

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

  - Speichern und schließen Sie die Datei.

- Wenn das System die Windows-Benutzerverwaltung zur Authentifizierung und den LDAP-Server zur Autorisierung verwendet, führen Sie die folgenden Teilschritte aus, um die Zuordnung von Benutzern zu Rollen aus LDAP zu ermitteln.

  **Hinweis**: Die Konfigurationsdatei UserRepository_Ldap.config enthält die serverspezifischen LDAP-Einstellungen. Passen Sie diese Datei an, bevor Sie die LDAP-Autorisierung aktivieren.

  - Pflegen Sie die Benutzer- und Gruppenzuordnungen in LDAP.
  - Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Core.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
  - Falls das Element `<serviceCredentials>` in der Datei vorhanden ist, entfernen Sie das Element oder kommentieren Sie es aus.

    **Beispiel**

    ```
    <behavior name="STSBehaviour">
         <!--  <serviceCredentials>
             <userNameAuthentication
    userNamePasswordValidationMode="Custom"
    customUserNamePasswordValidatorType="ModusSuite.Runtime.STS.L
    dapUserNamePasswordValidator,
    ModusSuite.Runtime.STSRuntimeService" />
          </serviceCredentials>  -->
       </behavior>
    ```

  - Speichern und schließen Sie die Datei.
  - Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **UserRepository.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
  - Setzen Sie den Wert des Attributs **userStore** auf **LDAP**.
  - Setzen Sie den Wert des Attributs **profileReadOption** auf **All**, **Role**, **None** oder **User**.

    **Beispiel**

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

  - Speichern und schließen Sie die Datei.

## Aktivieren der Windows-Authentifizierung

Um die Windows-Authentifizierung zu aktivieren, führen Sie die folgenden Schritte aus.

Diese Änderung hat Auswirkungen auf die folgenden Content Composer Anwendungen.

- Content Composer Studio bei Verwendung des Startparameters `/credentials:input`.
- Content Composer Client bei Verwendung des Startparameters `/credentials:input`.
- Content Composer Web Client

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Core.exe.config** mit einem Texteditor, der UTF-8 unterstützt.

2. Falls das Element `<serviceCredentials>` in der Datei vorhanden ist, entfernen Sie das Element oder kommentieren Sie es aus.

   **Beispiel**

```
<behavior name="STSBehaviour">
     <!--  <serviceCredentials>
         <userNameAuthentication
userNamePasswordValidationMode="Custom"
customUserNamePasswordValidatorType="ModusSuite.Runtime.STS.LdapUserN
amePasswordValidator, ModusSuite.Runtime.STSRuntimeService" />
      </serviceCredentials>  -->
   </behavior>
...
```

3. Speichern und schließen Sie die Datei.
4. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **UserRepository.config** mit einem Texteditor, der UTF-8 unterstützt.
5. Setzen Sie den Wert des Attributs `userStore` auf `Windows`.
6. Speichern und schließen Sie die Datei.
7. Starten Sie den Dienst **Composer.Core** neu.

## Konfigurieren der LDAP-Authentifizierung

Diese Konfigurationsvariante benutzt den LDAP-Server zur Authentifizierung und Autorisierung.

Die Konfigurationsdatei UserRepository_Ldap.config enthält die serverspezifischen LDAP-Einstellungen. Passen Sie diese Datei an, bevor Sie die LDAP-Autorisierung aktivieren.

Um die LDAP-Authentifizierung zu konfigurieren, führen Sie die folgenden Schritte aus:

1. Falls nötig, pflegen Sie die Benutzer- und Rollen-Attribute sowie Passwort und Gruppenzugehörigkeit in LDAP.
2. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Core.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
3. Suchen Sie die folgenden Zeilen:

```
<behaviors>
  <serviceBehaviors>
     <behavior name="STSBehaviour">
```

4. Fügen Sie innerhalb des Elements `<behavior name="STSBehaviour">` das folgende Element `<serviceCredentials>` hinzu.

```
<behavior name="STSBehaviour">
         <serviceCredentials>
            <userNameAuthentication
userNamePasswordValidationMode="Custom"
```

```
customUserNamePasswordValidatorType="ModusSuite.Runtime.STS.LdapUserN
amePasswordValidator, ModusSuite.Runtime.STSRuntimeService" />
        </serviceCredentials>
    </behavior>
```

5. Speichern und schließen Sie die Datei.
6. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **UserRepository.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
7. Setzen Sie den Wert des Attributs **userStore** auf **LDAP**.
8. Setzen Sie den Wert des Attributs **profileReadOption** auf **All**, **Role**, **None** oder **User**.
9. Speichern und schließen Sie die Datei.
10. Starten Sie den Dienst **Composer.Core** neu.
11. Um die Eingabe des LDAP-Benutzernamens sowie des Passworts beim Start von **Composer Studio** und **Composer Windows Client** zu ermöglichen, benutzen Sie den Kommandozeilenparameter `credentials:input`.

    **Beispiel**

    ```
    Composer.Studio.exe -credentials:input
    ```

    Weitere Informationen finden Sie unter Studio Startparameter.

**Result** LDAP führt die Benutzerauthentifizierung sowie die Ermittlung der Benutzergruppen durch. Das System ermittelt die Zuordnung von Benutzern zu Rollen aus LDAP. Falls nötig, liest das System die Benutzer und Rollenprofile aus LDAP.

## Anpassen der allgemeinen User Repository-Einstellungen

Um die User Repository-Einstellungen anzupassen, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **UserRepository.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Prüfen und ändern Sie, falls erforderlich, die Einstellungen.

| Attribut | Beschreibung |
|---|---|
| systemOId | Objekt ID des Systems, aus dem RoleMapper und UserProfile nach erfolgreicher Authentifizierung des Benutzers geladen werden. |
| roleMapper | Name des RoleMapper-Objekts, das verwendet wird, um die Gruppen den Rollen zuzuordnen.<br>Weitere Informationen finden Sie unter "Rollen-Mappings" in der *Content Composer Studio H*ilfe. |
| userProfile | Name des Profil-Objekts das verwendet wird, um die LDAP-Attribute den Content Composer Attributen zuzuordnen. |

| Attribut | Beschreibung |
|---|---|
| | Weitere Informationen finden Sie unter "Profile" in der *Content Composer Studio H*ilfe. |
| profileReadOption | Definiert ob Benutzerprofile für Benutzer und Rollen aus LDAP gelesen werden.<br><br>Diese Einstellung ist vom Parameter **userStore** abhängig.<br><br>So können z. B. die Benutzer aus der Windows-Benutzerverwaltung und die Profile aus LDAP gelesen werden.<br><br>**Hinweis**: Sind keine Benutzer- oder Rollenattribute definiert, setzen Sie den Wert auf **None**.<br><br>**Mögliche Werte**<br><br>• **None**: Es werden keine Benutzer- oder Rollenattribute aus LDAP gelesen. Das System verwendet die Defaultwerte.<br>• **User**: Die Benutzerattribute werden aus LDAP gelesen. Die Rollenattribute werden nicht aus LDAP gelesen, das System verwendet die Defaultwerte.<br><br>**Hinweis**: Für diese Einstellung wird eine gültige **UserRepository_Ldap.config** Datei benötigt.<br><br>• **Role**: Die Rollenattribute werden aus LDAP gelesen. Die Benutzerattribute werden nicht aus LDAP gelesen, das System verwendet die Defaultwerte.<br><br>**Hinweis**: Für diese Einstellung wird eine gültige **UserRepository_Ldap.config** Datei benötigt.<br><br>• **All**: Sowohl die Benutzer- als auch die Rollenattribute werden aus LDAP gelesen.<br><br>**Hinweis**: Für diese Einstellung wird eine gültige **UserRepository_Ldap.config** Datei benötigt. |
| userStore | Bestimmt, woher das System die Benutzer- und Gruppeninformationen bezieht.<br><br>**Mögliche Werte**<br><br>• **Windows**: Die Informationen werden aus der Windows-Benutzerverwaltung gelesen.<br>• **LDAP**: Die Informationen werden aus LDAP gelesen. |

Beispiel-Konfiguration

```
<?xml version="1.0" encoding="utf-8" ?>
   <userRepository
    systemOId ="dm"
    roleMapper = "Std_Mapping"
    userProfile = "Std_Profil"
    profileReadOption = "None"
    userStore = "Windows"
    />
```

## Anpassen der User-Repository-Einstellungen für LDAP

Um die LDAP-Einstellungen zu konfigurieren, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **UserRepository_LDAP.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Prüfen und ändern Sie, falls erforderlich, die Einstellungen.

| Attribut | Beschreibung |
|---|---|
| connectionString | Verbindungsinformationen zum LDAP-Server.<br>**Syntax**<br>`LDAP://Host:Port/BaseDN`<br>**Host**: LDAP-Server<br>**Port**: Standard: `389`, bei Verwendung von SSL: `636`<br>**BaseDN**: Root-Node in LDAP, unter dem die Benutzer- und Gruppen-Container abgelegt sind<br>**Beispiel**: `"LDAP://localhost:389/DC=DMS-PE,DC=DE"` |
| connectionProtection | **None**: Defaultwert<br>**SignAndSeal**: Die Verbindung wird gesichert, indem jedes Paket, das an den Server gesendet wird, digital signiert und verschlüsselt wird.<br>**SSL**: Die Verbindung wird über eine SSL-Verbindung hergestellt. |
| adminUser | Admin-Benutzer, der aus LDAP gelesen wird. Dieser Benutzer benötigt Lesezugriff auf alle Benutzer- und |

| Attribut | Beschreibung |
|---|---|
| | Gruppencontainer.<br>**Beispiel**: `= "CN=Administrator,CN=Users,DC=dms-pe,DC=de"` |
| adminPassword | Mit dem Tool **Encoder.exe** verschlüsseltes Passwort dieses Benutzers. |
| groupContainerDN | Name des Containers, in dem die Gruppen in LDAP definiert sind.<br>Ist dieser Parameter nicht leer, wird mit der Suche nach der Gruppe in diesem Container begonnen.<br>Ist dieser Parameter leer, wird mit der Suche in dem im Parameter **ConnectionString**angegebenen BaseDN begonnen.<br>Es wird empfohlen, diesen Parameter leer zu lassen (""). |
| userContainerDN | Name des Containers, in dem die Benutzer in LDAP definiert sind.<br>Ist dieser Parameter nicht leer, wird mit der Suche nach dem Benutzer in diesem Container begonnen.<br>Ist dieser Parameter leer, wird mit der Suche in dem im Parameter **ConnectionString**angegebenen BaseDN begonnen.<br>Es wird empfohlen, diesen Parameter leer zu lassen (""). |
| groupFilter | Suchfilter, der die LDAP-Klasse für die Gruppen identifiziert.<br>**Beispiel für Microsoft Active Directory:**<br>`(&(objectCategory=group)(objectClass=group){0})`<br>Die LDAP-Klasse für Gruppen hat die Attribute `objectCategory=group` und `objectClass=group`.<br>{0} wird automatisch durch den im Element `AttributeMapGroupName` angegebenen Gruppennamen ersetzt, wie z. B. `(member=CN=Test User,CN=Users,DC=dms-pe,DC=de)`. |

| Attribut | Beschreibung |
|---|---|
| | Um die Gruppe **Admin** zu finden, kann also der folgende Filter verwendet werden:<br>`(&(objectCategory=group)(objectClass=group)(member=Admin)`<br>Hat die LDAP-Klasse z. B. nur das Attribut `objectClass`, sieht die Definition wir folgt aus:<br>`(&(objectClass=group){0})`<br>**Hinweis**: Das Zeichen "&" muss in der Config-Datei durch & ersetzt werden. |
| userFilter | Suchfilter, der die LDAP-Klasse für die Benutzer identifiziert.<br>**Beispiel für Microsoft Active Directory:**<br>`(&(objectCategory=person)(objectClass=user){0})`<br>Die LDAP-Klasse für Benutzer hat die Attribute `objectCategory=person` und `objectClass=user`.<br>{0} wird automatisch durch den im Element `AttributeMapUserName` angegebenen Benutzernamen ersetzt, wie z. B. `(sAMAccountName=Meier)`.<br>Um den Benutzer **Meier** zu finden, kann also der folgende Filter verwendet werden:<br>`(&(objectCategory=person)(objectClass=person)(sAMAccountName=Meier)`<br>Hat die LDAP-Klasse z. B. nur das Attribut objectClass, sieht die Definition wie folgt aus:<br>`(&(objectClass=person){0})`<br>**Hinweis**: Das Zeichen "&" muss in der Config-Datei durch & ersetzt werden. |
| attributeMapGroupName | Attribut, das verwendet wird, um die Gruppen zu identifizieren, zu der der Benutzer gehört, wie z. B. "member". |
| attributeMapUserName | Attribut, das verwendet wird, um einen Benutzer zu identifizieren, wie z. B. "sAMAccountName". |

| Attribut | Beschreibung |
|---|---|
| includeLocalGroups | **TRUE**: Zu den globalen Gruppen werden auch die lokalen Gruppen ermittelt.<br>**FALSE**: Die lokalen Gruppen werden nicht ermittelt. |

#### Beispiel-Konfiguration

```
<?xml version="1.0" encoding="utf-8" ?>
   <userRepository_Ldap
      connectionString = "LDAP://localhost:389/DC=DMS-PE,DC=DE"
      connectionProtection = "None"
      adminUser = "CN=Administrator,CN=Users,DC=dms-pe,DC=de"
      adminPassword = "password
      groupContainerDN = ""
      userContainerDN = ""
      groupFilter = "(&amp;(objectCategory=group)(objectClass=group)
{0})"
      userFilter = "(&amp;(objectCategory=person)(objectClass=user){0})"
      attributeMapGroupName = "member"
      attributeMapUserName = "sAMAccountName"
      includeLocalGroups = "TRUE"
  />
```

# Verfügbare Standardprozess

## Der Prozess MWS Cleanup

### Informationen zum MWS Cleanup-Prozess

Der Prozess **MWS_Standard_Cleanup** ist ein vordefinierter Prozess, mit dem Sie die folgenden Daten aus der MWS-Datenbank und dem Dateisystem löschen können.

- MWS-Sessions, die älter als die Anzahl der in der Skript-Variablen `sessionAgeInDays` definierten Tage sind.

  Die Default-Einstellung ist 30.

- Beendete MWS-Prozesse.

Der MWS Cleanup-Prozess besteht aus folgenden Objekten:

- Dem Prozess **MWS_Standard_Cleanup**.
- Den Skripten **MWS_Standard_Cleanup** und **MWS_Standard_Cleanup_ErrorHandler**.

Weitere Informationen finden Sie im Kapitel "Informationen zum MwsAssistantManager" in der *Content Composer Studio H*ilfe.

## Importieren des MWS Cleanup-Prozesses

Um den **MWS_Standard_Cleanup**-Prozess zu importieren, führen Sie die folgenden Schritte im Content Composer Studio durch.

1. Klicken Sie im Reiter **Administration** auf die Schaltfläche **Repository Import** .
2. Geben Sie im Feld **Angabe des Importverzeichnisses** den Text **mws_cleanup** ein.
3. Klicken Sie im Reiter **Repository Import** auf die Schaltfläche **Ausführen** .
4. Öffnen Sie im **Navigator** den Prozess **MWS_Standard_Cleanup**.
5. Selektieren Sie im oberen Bereich des **Bearbeitungsfensters** das **Timer**-Objekt und passen Sie falls nötig im **Objektinspektor** die Eigenschaften an.
6. Wählen Sie im unteren Editor-Fenster die Komponente **MwsAssistantManager** und dann im **Objektinspektor** den **DB-Alias**, der die Verbindungsinformationen zur MWS-Datenbank enthält.
7. Optional. Ändern Sie die Eigenschaft **Bedingung**.

   Der vordefinierte Wert `MWS_STATUS=2` löscht alle MWS-Prozesse mit dem Status **Finished**.

   **Beispiel**

   Um alle Prozesse zu löschen, die seit mindestens 5 Tagen abgeschlossen sind, ändern Sie die Bedingung wie folgt:

   **Für SQL Server**: `MWS_STATUS = 2 and DATEDIFF(day, MWS_LASTSAVED, CURRENT_TIMESTAMP) < 6`

   **Für Oracle**: `MWS_STATUS = 2 and (SYSDATE-mws_lastsaved < 6)`
8. Optional. Um die Anzahl der Tage, nach denen MWS-Sessions durch den **MWS_Standard_Cleanup**-Prozess gelöscht werden zu ändern, führen Sie die folgenden Teilschritte aus.
   1. Wählen Sie im oberen Bereich des **Editors** das Objekt **MWS_Cleanup** aus.
   2. Öffnen Sie vom **Objektinspektor** das Skript **MWS_Standard_Cleanup**.
   3. Suchen Sie im Skript die Variable `sessionAgeInDays` und ändern Sie ihren Wert wie erforderlich.

      **Beispiel**

      ```
      sessionAgeInDays = 20
      ```
9. Speichern Sie die Änderungen.
10. Klicken Sie im Reiter **Prozessdesigner Tools** auf die Schaltfläche **Kompilieren** .
11. Konfigurieren Sie einen Content Composer Windows-Dienst als Host für den **MWS_Standard_**

**Cleanup**-Prozess.

Weitere Informationen finden Sie unter Prozesse hosten.

# Der Prozess Odin Cleanup

## Informationen zum Odin Cleanup-Prozess

Der Prozess **Odin_Standard_Cleanup** ist ein vordefinierter Prozess, mit dem Sie die folgenden Objekte löschen können:

- Odin-Prozesse ohne zugeordnete Jobs.
- Einen beendeten oder zu löschenden Odin-Stapel.

Weitere Informationen finden Sie unter "Informationen zu OdinDbTasks" in der *Content Composer Studio Hilfe*.

## Importieren des Odin Cleanup-Prozesses

Um den **Odin_Standard_Cleanup**-Prozess zu importieren, führen Sie die folgenden Schritte im Content Composer Studio durch.

Das im Prozess verwendete Skript **Odin_Standard_Cleanup** ist lauffähig und benötigt keine Anpassungen.

1. Klicken Sie im Reiter **Administration** auf die Schaltfläche **Repository Import** .
2. Geben Sie im Feld **Angabe des Importverzeichnisses** den Text `odin_cleanup` ein.
3. Klicken Sie im Reiter **Repository Import** auf die Schaltfläche **Ausführen** .
4. Öffnen Sie im **Navigator** den Prozess **Odin_Standard_Cleanup**.
5. Selektieren Sie im oberen Bereich des **Bearbeitungsfensters** das **Timer**-Objekt und passen Sie falls nötig im **Objektinspektor** die Eigenschaften an.
6. Selektieren Sie im unteren Bereich des **Bearbeitungsfensters** die Komponente **OdinDbTasks** und wählen sie dann im **Objektinspektor** den benötigten **DB-Alias** aus.
7. Speichern Sie die Änderungen.
8. Klicken Sie im Reiter **Prozessdesigner Tools** auf die Schaltfläche **Kompilieren** .

# Der Prozess OdinOnBaseArchive

## Informationen zum OdinOnBaseArchive-Prozess

Der Prozess **OdinOnBaseArchive_Standard** ist ein vordefinierter Prozess, mit dem Sie Dokumente, die zur Archivierung bereit sind, für den Import in das OnBase-Archiv vorbereiten können.

Weitere Informationen finden Sie unter "Informationen zu OdinOnBaseArchive" in der *Content Composer Studio Hilfe*.

### Importieren der OdinOnBaseArchive-Prozesses

Um den **OdinOnBaseArchive_Standard**-Prozess zu importieren, führen Sie die folgenden Schritte im Content Composer Studio durch.

1. Klicken Sie im Reiter **Administration** auf die Schaltfläche **Repository Import** .
2. Geben Sie im Feld **Angabe des Importverzeichnisses** den Text `OdinOnbaseArchive_proc` ein.
3. Klicken Sie im Reiter **Repository Import** auf die Schaltfläche **Ausführen** .
4. Öffnen Sie im **Navigator** den Prozess **OdinOnBaseArchive_Standard**.
5. Selektieren Sie im oberen Bereich des **Bearbeitungsfensters** das **Timer**-Objekt und passen Sie falls nötig im **Objektinspektor** die Eigenschaften an.
6. Selektieren Sie im unteren Bereich des **Bearbeitungsfensters** die Komponente **OdinOnBaseArchive** und wählen sie dann im **Objektinspektor** den benötigten **DB-Alias** aus.
7. Wählen Sie im **Objektinspektor** die Eigenschaft **HotFolderDirectory** und geben Sie das Verzeichnis ein oder wechseln Sie zu dem Verzeichnis, in dem Sie die archivierten Dokumente speichern möchten.
8. Selektieren Sie die Eigenschaft **StackSelect** und geben Sie die gewünschte SQL-Anweisung an.
9. Optional. Selektieren Sie die Eigenschaft **TraceFileName** und geben Sie die **Compart** Trace-Datei ein oder navigieren Sie zu der Datei.
10. Führen Sie einen der folgenden Teilschritte durch.
    - Wenn Sie eine **Microsoft SQL Server** Datenbank verwenden, benennen Sie die Lookup-Tabelle **OdinSQL_MS_SqlServer** in **OdinSQL** um.
    - Wenn Sie eine **Oracle** Datenbank verwenden, benennen Sie die Lookup-Tabelle **OdinSQL_Oracle** in **OdinSQL** um.
11. Speichern Sie die Änderungen.
12. Klicken Sie im Reiter **Prozessdesigner Tools** auf die Schaltfläche **Kompilieren** .

## Der Prozess ScheduledTransfers

### Informationen zum ScheduledTransfers-Prozess

Der Prozess **ScheduledTransfers** ist ein vordefinierter Prozess, mit dem Sie vordefinierte Transfer-Jobs ausführen können.

Weitere Informationen finden Sie unter "Informationen zu Transferjobs" und "Über Transfer" in der *Content Composer Studio Hilfe*.

### Importieren des ScheduledTransfer-Prozesses

Um den **ScheduledTransfers**-Prozess zu importieren, führen Sie die folgenden Schritte im Content Composer Studio durch.

1. Klicken Sie im Reiter **Administration** auf die Schaltfläche **Repository Import** .
2. Geben Sie im Feld **Angabe des Importverzeichnisses** den Text `lifecycle_proc` ein.
3. Klicken Sie im Reiter **Repository Import** auf die Schaltfläche **Ausführen** .
4. Öffnen Sie im **Navigator** den Prozess **ScheduledTransfers**.
5. Selektieren Sie im oberen Bereich des **Bearbeitungsfensters** das **Timer**-Objekt und passen Sie falls nötig im **Objektinspektor** die Eigenschaften an.
6. Selektieren Sie im unteren Bereich des **Bearbeitungsfensters** die Komponente **Transfer** und wählen sie dann im **Objektinspektor** den benötigten **DB-Alias** aus.
7. Speichern Sie die Änderungen.
8. Klicken Sie im Reiter **Prozessdesigner Tools** auf die Schaltfläche **Kompilieren** .

# Prozesse hosten

## Installation eines Windows-Services

Zur Installation eines Windows-Dienstes zum Hosten von Content Composer-Prozessen führen Sie die Schritte in den folgenden Kapiteln durch.

1. Kopieren Sie die erforderlichen Dateien. Weitere Informationen finden Sie im Kapitel Kopieren der erforderlichen Dateien.
2. Passen Sie die Konfigurationsdateien an. Weitere Informationen finden Sie im Kapitel Anpassen der neu erstellten Konfigurationsdatei.
3. Legen Sie die Prozesse fest. Weitere Informationen finden Sie im Kapitel Festlegen der Prozesse.
4. Registrieren Sie den Dienst. Weitere Informationen finden Sie im Kapitel Registrieren der Dienste.

### Kopieren der erforderlichen Dateien

Um die Dateien zu kopieren, führen Sie die folgenden Schritte durch.

1. Erzeugen Sie im Verzeichnis **%Composerdir%** eine Kopie der Datei **Composer.WindowsServiceHost.exe**.
2. Geben Sie der kopierten Datei einen aussagekräftigen Namen.
3. Erstellen Sie im Verzeichnis **%Composerdir%** eine Kopie der CONFIG-Datei **Composer.WindowsServiceHost.exe.config**.
4. Geben Sie der kopierten CONFIG-Datei denselben Namen wie die EXE-Datei um, mit Ausnahme der Dateiendung **.config**.

   **Beispiel**

- Kopieren Sie die Datei **Composer.WindowsServiceHost.exe** nach **BatchServiceHost.exe**.
- Kopieren Sie die Datei **Composer.WindowsServiceHost.exe.config** nach **BatchServiceHost.exe.config**.

### Anpassen der neu erstellten Konfigurationsdatei

Um die neue Konfigurationsdatei anzupassen, führen Sie die folgenden Teilschritte durch.

1. Öffnen Sie die in Kopieren der erforderlichen Dateien neu erstellte CONFIG-Datei mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die Zeile, die wie folgt beginnt:
   ```
   <add name="tracefile_listener" type=
   ```
3. Ändern Sie den Wert des Attributs `initializeData` auf den Dateiname der Log-Datei.
   **Beispiel**
   ```
   <add name="tracefile_listener"
   type="ModusSuite.Common.SystemFramework.DailyTraceListener, ModusSu
   ite.Common.SystemFramework" delimiter=";"
   initializeData="BatchServiceHost_log.txt"
   traceOutputOptions="DateTime"/>
   ```
4. Falls der Prozess Dokumente generiert, führen Sie die folgenden Teilschritte durch, um das Microsoft Word Template zuzuweisen.
   1. Suchen Sie nach dem Knoten `<appSettings>`.
   2. Falls die **Normal.dotm** im Installationsverzeichnis liegt, fügen Sie die folgenden Zeile ein:
      ```
      <add key="wordtemplate" value="%Composerdir%Normal.dotm" />
      ```
      **Hinweis**: Die Installation erstellt eine Umgebungsvariable namens **Composerdir**, die den Pfad zum Installationsverzeichnis des Content Composer Studios enthält.
   3. Wenn sich die Normal.dotm nicht im Installationsverzeichnis befindet, fügen Sie die folgende Zeile hinzu und ersetzen Sie dabei `PathToNormal.dotm` durch den Pfad der benötigten Normal.dotm.
      ```
      <add key="wordtemplate"
      value="PathToNormal.dotm\Normal.dotm" />
      ```
5. Speichern und schließen Sie die Datei.

### Festlegen der Prozesse

Um die gewünschten Prozesse in der Konfigurationsdatei einzutragen, führen Sie einen oder beide der folgenden Schritte durch.

**Hinweis**: Sind beide Varianten konfiguriert, so werden zunächst die Einzelprozesse gestartet und anschließend die über Aliastabellen referenzierten Prozesse.

- Um einen Einzelprozess zu konfigurieren, führen Sie die folgenden Teileschritte durch.
  - Suchen Sie in der Konfigurationsdatei die folgenden Zeilen und entfernen Sie die Kommentarzeichen.

    ```
    <processes>
    <!-- each process-element represents a process that will be
    started -->
    <!--
      <process name="ModBatch_XmlFile" reference="ModBatch_
    XmlFile|29012008-13-prc-modb|2048|Process|None|modstd_
    megatron"/>
      -->
    </processes>
    ```

  - Ändern Sie das Attribut `name` auf den Namen des Prozesses, den Sie konfigurieren wollen.
    **Beispiel** `<process name="`**`Process_ConvertMultipleFiles`**`" reference="ModBatch_XmlFile|29012008-13-prc-modb|2048|Process|None|modstd_megatron"/>`

    **Hinweis**: Um mehr als einen Prozess zu konfigurieren, duplizieren Sie die Zeile.

  - Starten Sie das **Content Composer Studio**.
  - Rechtsklicken Sie im im unteren **Navigatorfenster** auf den Prozess und klicken Sie dann auf **Objektinformationen anzeigen**.
  - Kopieren Sie dann im **Ausgabefenster** den Text, der nach **SystemObjectID** angezeigt wird:
    **Beispiel**
    `RepositoryReference:`
    `Name, ObjectID, database ID, objectType, referenceType,`
    `SystemObjectID:` **`Process_ConvertMultipleFiles|20112009-432-prc-dm|985|Process|None|dm`**
  - Fügen Sie den Text in der Konfigurationsdatei als Inhalt des reference-Attributs ein.
    **Beispiel**
    `<process name="Process_ConvertMultipleFiles"`
    `reference="`**`Process_ConvertMultipleFiles|20112009-432-prc-dm|985|Process|None|dm`**`"/>.`

- Optional. Wenn Sie den gleichen Prozess mehrmals ausführen möchten, duplizieren Sie die entsprechende Zeile und ändern Sie dann das Attribut `name`.

  **Beispiel**

  ```
  <processes>
  <process name="ODIN_Rendition 1" reference="Odin_ Rendition|
  19022011-10-prc-odv|2233|Process|None|modstd_ megatron" />
  <process name="ODIN_Rendition 2" reference="Odin_ Rendition|
  19022011-10-prc-odv|2233|Process|None|modstd_ megatron" />
  </processes>
  ```

- Um Prozesse zu konfigurieren, die durch eine Aliastabelle referenziert werden, führen Sie die folgenden Teilschritte durch.
  - Suchen Sie in der Konfigurationsdatei die folgenden Zeilen und entfernen Sie die Kommentarzeichen.

    ```
    <aliastables>
    <!--
      <aliastable name="batchProcesses"
    systemoid="mod5processes"/>
      -->
    </aliastables>
    ```

  - Ändern Sie das Attribut `name` auf den Namen der Aliastabelle.
  - Ändern Sie das Attribut `systemoid` auf die ID des Systems, in dem die Aliastabelle liegt.

    **Hinweise**

    Wenn Sie mehr als eine Aliastabelle konfigurieren, stellen Sie sicher, dass keine identischen Aliase erstellt werden, da dies zu einem Fehler führen würde.

    Da ein Alias auch mehrere Verweise auf unterschiedliche Objekte beinhalten kann, gelten folgende Regeln:

    - Es werden nur Verweise auf normale Prozesse (keine MWS Prozesse oder andere Typen) verarbeitet
    - Beinhaltet ein Schlüssel mehrere Prozesse, so bekommt der erste als Instanz-Id den Schlüsselnamen und die folgenden den Schlüsselnamen plus einen mit eins beginnenden fortlaufenden Zähler z.B. **OdinOnline 1**.

## Registrieren der Dienste

Um den Service in den Windows-Diensten zu registrieren, führen Sie die folgenden Schritte durch.

1. Öffnen Sie als Administrator ein CMD-Fenster und wechseln Sie in das Verzeichnis **%Composerdir%**.
2. Führen Sie folgenden Befehl aus und ersetzen Sie dabei die Platzhalter durch die erforderlichen

Parameter.

```
Install_Composer_service "[ServiceName]" "[DisplayName]" "
[ImageName]" "[Description]"
```

| Parameter | Beschreibung |
|---|---|
| **ServiceName** | Beschreibung des Dienstes in der Dienstverwaltung |
| **DisplayName** | Anzeigename dieser Service-Instanz in der Dienstverwaltung |
| **ImageName** | Name der EXE-Datei |
| **Beschreibung** | Beschreibung des Dienstes in der Dienstverwaltung |

**Beispiel**

```
Install_Composer_service "BatchService" "BatchServiceComposer"
"BatchServiceHost.exe" "Composer Tools Service"
```

## Anpassen der Konfigurationsdateien

Um die Konfiguration abzuschließen, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Studio.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die Zeilen, die wie folgt beginnen:

   ```
   <endpoint name=
   ```

3. Duplizieren Sie eines der vorhandenen `endpoint` Elemente.
4. Ändern Sie das Attribut `name` auf den Namen der Prozesses.
5. Setzen Sie im Attribut `address` die Portnummer auf einen Port, der nicht bereits von einem anderen Endpunkt verwendet wird.

   **Beispiel**

   ```
   <endpoint name="CoCoProcess_SVC"
         address="http://localhost:4724/RemoteControlService"
         binding="wsDualHttpBinding"
         bindingConfiguration="Binding1"

   contract="ComposerSuite.Runtime.RemoteControlContract.IRuntimeControlS
   ervice" />
   ```

6. Speichern und schließen Sie die Datei.
7. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **RemoteControl.config** mit einem Texteditor, der UTF-8 unterstützt.
8. Suchen Sie die Zeilen, die wie folgt beginnen:

```
<service name=
```

9. Duplizieren Sie eines der vorhandenen `service` Elemente.
10. Setzen Sie das Attribut `name` auf den Namen des Endpunkts, den Sie soeben in der Datei **Composer.Studio.exe.config** in den obigen Schritten erstellt haben.
11. Setzen Sie das Attribut `display` innerhalb des Elements `service` auf einen aussagekräftigen Namen.

    **Beispiel**

    ```
    <service name="CoCoProcess_SVC" display="Composer Process Service"
    runtime="TRUE" endpoint="RemoteControlService"/>
    ```

    Das folgende Beispiel überwacht einen lokalen Computer mit fünf Diensten.

    ```
    <remotecontrol>
       <computers>
          <computer name="." display="cocal computer">
             <services>
                <service name="Composer.Core.SVC" display="Core"
    runtime="true" endpoint="RemoteControlServiceCore"/>
                <service name="Composer.MWS.SVC" display="MWS"
    runtime="true" endpoint="RemoteControlServiceMWS"/>
                <service name="Composer.XWS.SVC" display="XWS"
    runtime="true" endpoint="RemoteControlServiceXWS"/>
                <service name="Composer.OWS.SVC" display="OWS"
    runtime="true" endpoint="RemoteControlServiceOWS"/>
                <service name="CoCoProcess_SVC" display="Composer Process
    Service" runtime="true" endpoint="RemoteControlService"/>
             </services>
          </computer>
       </computers>
    </remotecontrol>
    ```

| Attribut | Beschreibung |
|---|---|
| **name** | Anzeigename des Dienstes in den Windows-Diensten. |
| **display** | Der in den Windows-Diensten angezeigte Text. |
| **runtime** | **Mögliche Werte**:<br>**True**: Runtime-Dienst<br>**False**: Windows Dienst<br>Sie können einen Runtime-Dienst auch mit der Einstellung FALSE |

| Attribut | Beschreibung |
|---|---|
| | konfigurieren, wenn die darin laufenden Prozesse nicht angezeigt werden sollen.<br>Dies ist z.B. dann sinnvoll, wenn eine Runtime ausschließlich Webdienste hostet und keine Prozesse ausführen soll. |
| **endpoint** | Der Endpoint für die WCF-Kommunikation, wenn es sich um einen Runtime-Dienst handelt. |

12. Speichern und schließen Sie die Datei.
13. Öffnen Sie die in Kopieren der erforderlichen Dateien neu erstellte CONFIG-Datei mit einem Texteditor, der UTF-8 unterstützt.
14. Suchen Sie die Zeilen, die wie folgt beginnen:

    ```
    <remotecontrol address="http:
    ```

15. Tragen Sie dort die Portnummer der endpoint-Adresse ein, den Sie in der **Composer.Studio.exe.config** vergeben haben.
    **Beispiel** `<remotecontrol address="http://127.0.0.1:4724/RemoteControlService" />`
16. Suchen Sie in der Datei die folgende Zeile

    ```
    <service name="ComposerSuite.Runtime.RemoteControlService">
    ```

17. Setzen Sie die Portnummer auf die gleiche Portnummer wie im vorherigen Schritt.

    **Beispiel**

    ```
    <services>
          <service name="ComposerSuite.Runtime.RemoteControlService">
             <endpoint
             address="http://127.0.0.1:4724/RemoteControlService"
             binding="wsDualHttpBinding"
             bindingConfiguration ="RemoteControlBinding"

    contract="ComposerSuite.Runtime.RemoteControlContract.IRuntimeControlS
    ervice"/>
    ```

18. Speichern und schließen Sie die Datei.

## Deinstallation eines Windows-Dienstes

Um einen Windows-Dienst zu deinstallieren, führen Sie den folgenden Schritt aus.

1. Öffnen Sie als Administrator ein CMD-Fenster und wechseln Sie in das Verzeichnis **%Composerdir%**.

2. Führen Sie folgenden Befehl aus und ersetzen Sie dabei `[ServiceName]` durch den Namen des Dienstes, den Sie deinstallieren möchten.

   ```
   Uninstall_Composer_service.cmd [ServiceName]
   ```

   **Beispiel**

   ```
   Uninstall_Composer_service.cmd BatchService
   ```

# Datenbanken für Volltextsuche einrichten

## Oracle

Der benötigte Index **Oracle Text** ist Teil der Oracle Standardkonfiguration.

Je nach dem verwendeten Betriebssystem kann es notwendig sein, eine entsprechende XPS Erweiterung hinzuzufügen.

## SQL Server

Stellen Sie die Verfügbarkeit der Volltextsuche (Index **SQL Full Text**) für Ihr System sicher.

Informationen zur Volltextsuche unter MS SQL Server finden Sie hier:

http://msdn.microsoft.com/en-us/library/ms403375%28SQL.90%29.aspx

http://technet.microsoft.com/en-us/library/ms142497.aspx

http://technet.microsoft.com/en-us/library/ms142499.aspx

**IFilters**

Die Suche in Binärdateien erfordert IFilter, z.B. für .PDF und MS Office Dateien:

| | |
|---|---|
| Adobe PDF IFilter 9 (64-Bit) | **Link** : ftp://ftp.adobe.com/pub/adobe/acrobat/win/9.x/<br>**Dateiname**: PDFiFilter64Installer.zip |
| Microsoft Office 2010 Filter Packs | http://www.microsoft.com/download/en/details.aspx?id=17062 |
| Microsoft XPS Viewer (Windows 7, Windows 2008 Server) | Stellen Sie sicher, dass der Microsoft XPS Viewer installiert oder in Windows 2008 R2 das entsprechende Feature aktiviert ist. |

1. Laden Sie die erforderlichen Filter herunter und installieren Sie sie.
2. Führen Sie den folgenden Code auf Ihrem SQL Server aus:

```
exec sp_fulltext_service 'load_os_resources', 1;
exec sp_fulltext_service 'verify_signature', 0;
go
```

3. Starten Sie den SQL Server neu und überprüfen Sie anschließend, ob die IFilter installiert sind (beispielsweise .PDF and .DOCX):

```
exec sp_help_fulltext_system_components 'filter';
```

**XPS Erweiterung**

Je nach dem verwendeten Betriebssystem kann es notwendig sein, eine entsprechende XPS Erweiterung hinzuzufügen.

# Importieren eines Transferjobs

Um einen zuvor exportierten Transferjob in ein Content Composer-System zu importieren, führen Sie die folgenden Schritte aus.

Weitere Informationen finden Sie unter "Transferjobs" in der *Content Composer Studio H*ilfe.

1. Öffnen Sie auf dem Content Composer Server als Administrator ein CMD-Fenster und wechseln Sie in das Verzeichnis **%Composerdir%**.
2. Führen Sie folgenden Befehl aus und ersetzen Sie dabei die Platzhalter durch die erforderlichen Parameter.

```
TransferConsole.exe -SysOId:[SystemObjectId] -Dir:[Directory] -
Action:import -SetReadOnlyFlag:[yes/no]
```

| Parameter | Beschreibung |
|---|---|
| **SysOId** | Objekt-Id des Content Composer Systems, in das der Transferjob importiert wird. |
| **Dir** | Verzeichnis im Dateisystem. |
| **Action** | Auszuführende Aktion<br>**import**: Importiert den Transferjob aus dem angegebenen Verzeichnis in das angegebene System. |
| **SetReadOnlyFlag** | **Mögliche Werte**<br>**Yes**: Setzt den Schreibschutz für alle importierten Objekte auf `true`.<br>**No**: Setzt den Schreibschutz für alle importierten Objekte auf `false`. |

**Beispiel**

Der folgende Befehl importiert den Transferjob aus dem Unterverzeichnis `Invoices` in das Content Composer System mit der System-Objekt-Id `ProdSystem` und setzt den Schreibschutz für alle importierten Objekte auf `True` gesetzt.

```
TransferConsole.exe -SysOId:ProdSystem -Dir:Invoices -Action:import -
SetReadOnlyFlag:yes
```

# Informationen zur System-Zugriffssteuerung

Wenn die Systemzugriffskontrolle aktiviert ist, wird der Zugriff auf jedes System verweigert, für das die Rolle den Zugriff nicht ausdrücklich erlaubt.

Es ist daher zwingend erforderlich. Systemzugangsrechte im **Content Composer Studio** zu vergeben, bevor Sie die Systemzugriffssteuerung aktivieren.

**Zugriffsrechte für einen Content Composer Windows-Dienst**

Das Windows Benutzerkonto, mit dem der Windows-Dienst ausgeführt wird, benötigt die folgenden Zugriffsberechtigungen:

- Zugriffsberechtigung für das Content Composer System, welches den auszuführenden Content Composer Prozess enthält.
- Zugriffsberechtigungen für das Quell- und Zielsystem der ausgeführten Transfers, wenn im auszuführenden Content Composer Prozess die Komponente **Transfer** verwendet wird.

**Zugriffsrechte für einen Content Composer Client-Benutzer**

- **Content Composer Client** Benutzer benötigen Zugriffsrechte zu dem System, das den vom MWS-Server verwendeten **MWS**-Prozess enthält.

  Weitere Informationen finden Sie unter Das mws-Element in Composer.MWS.exe.config.

- Werden die Odin-Ansichten im **Content Composer Client** verwendet, benötigen die Benutzer Zugriffsrechte zu dem System, das die Abfragedefinitionen der Odin-Ansichten enthält.

  Weitere Informationen finden Sie unter Konfigurieren der Odin-Ansichten für den Client.

## Aktivieren der System-Zugriffssteuerung

Um diese System-Zugriffssteuerung zu aktivieren, führen Sie die folgenden Schritte durch.

**Prerequisite**

Konfigurieren Sie die Systemzugriffsrechte in Content Composer Studio, bevor Sie die System-Zugriffssteuerung aktivieren.

Weitere Informationen finden Sie unter "Rollen" in der *Content Composer Studio H*ilfe.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Core.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die Zeile, die wie folgt beginnt:

```
<add key="rws_enable_access_check" value=
```

3. Setzen Sie den Wert des `value` Attributs auf `1`.

   **Beispiel**

   ```
   <add key="rws_enable_access_check" value="1"/>
   ```

   **Hinweis**: Um diese System-Zugriffssteuerung zu deaktivieren, setzen Sie den Wert des `value`-Attributs auf `0`.

4. Speichern und schließen Sie die Datei.

# Informationen zum Cache Management

Das Cache-Management ermöglicht es, Objekte automatisch aus dem Cache zu entfernen.

**In folgenden Fällen entfernt das Cache-Management Objekte aus dem Cache:**

- Der Benutzer ändert und speichert ein Objekt im Content Composer Studio.
- Nach erfolgreicher Ausführung eines Transfers
- Nach erfolgreicher Rücknahme eines Transfers

## Aktivieren des Cache Managements

Um das Cache Management zu aktivieren, führen Sie die folgenden Schritte aus.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Core.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie nach dem `<appSettings>` Element.
3. Wenn die folgenden Einträge nicht vorhanden sind, fügen Sie sie ein. Sind die Einträge auskommentiert, entfernen Sie die Kommentarzeichen.

   ```
   <appSettings>
   ...
      <add key="rwsMulticastIP" value="224.100.0.1" />
      <add key="rwsMulticastPort" value="9050" />
      <add key="rwsMulticastTTL" value="50" />
      <add key="rws_sendudpmessage" value="y" />
   ...
   </appSettings>
   ```

4. Ändern Sie die Werte bei Bedarf.

| Key | Beschreibung |
|---|---|
| rwsMulticastIP | IP-Adresse der Multicast-Gruppe<br>Die IP muss im Bereich 224.0.0.0 - 239.255.255.255 liegen.<br>**Hinweis**: |

| Key | Beschreibung |
|---|---|
| | • Die Router müssen UDP Multicasting unterstützen.<br>• Fehlt der Eintrag, werden die Objekte nicht aus dem Cache entfernt. |
| rwsMulticastPort | Der zu verwendende Port. |
| rwsMulticastTTL | Definiert die Time To Live (TTL). TTL definiert die maximale Anzahl von Routern, die ein Paket durchlaufen kann bevor der Router das Paket löscht. |
| rws_ sendudpmessage | **y**: Das Cache Management wird aktiviert.<br>**n**: Das Cache Management wird nicht aktiviert. |

5. Speichern und schließen Sie die Datei.
6. Öffnen Sie die Datei **Composer.Studio.exe.config** mit einem Texteditor, der UTF-8 unterstützt und führen Sie dann die folgenden Teilschritte aus.
    1. Suchen Sie nach dem `<appSettings>` Element.
    2. Wenn die folgenden Einträge nicht vorhanden sind, fügen Sie sie ein. Sind die Einträge auskommentiert, entfernen Sie die Kommentarzeichen.

```
<appSettings>
...
   <add key="rwsMulticastIP" value="224.100.0.1" />
   <add key="rwsMulticastPort" value="9050" />
   <add key="rwsMulticastTTL" value="50" />
...
</appSettings>
```

    3. Ändern Sie die Werte bei Bedarf.
    4. Speichern und schließen Sie die Datei.
    5. Wiederholen Sie die vorherigen Schritte, um den Cache der folgenden Dienste und Tools zu leeren, indem Sie die jeweilige CONFIG-Datei ändern.

| Dienst / Tool | CONFIG Datei |
|---|---|
| MWS Dienst | Composer.MWS.exe.config |
| OWS Dienst | Composer.OWS.exe.config |
| XWS Dienst | Composer.XWS.exe.config |

| Dienst / Tool | CONFIG Datei |
|---|---|
| DocXCompiler | DocXCompiler.exe.config |
| Composer.Tool | Composer.Tool.exe.config |

# Informationen zur OnBase Document REST API

Dieser Abschnitt beschreibt die Konfigurationseinstellungen, die zum Archivieren von Dokumenten in einem OnBase-Archiv unter Verwendung der OnBase Document REST API erforderlich sind.

Führen Sie die Schritte in den Kapiteln Verbindung zur OnBase Document REST API und Anpassen der IdP-Konfigurationsdatei aus.

## Verbindung zur OnBase Document REST API

Damit Content Composer Studio eine Verbindung zur OnBase Document API herstellen und die vorhandenen OnBase Dokumenttypen und Schlüsselworte abrufen kann, führen Sie die folgenden Schritte aus:

**Hinweis**: Verwenden Sie das Tool **Encoder.exe** aus dem Verzeichnis **%Composerdir%**, um die Werte für `idpPassword` und `idpClientSecret` zu verschlüsseln.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Studio.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Falls noch nicht vorhanden, fügen Sie die folgenden Zeilen innerhalb des `<appsettings>` Elements ein:

```
<add key="idpClientId" value="TestClientID" />
<add key="idpUserName" value="TestUserName" />
<add key="idpPassword"
value="jZhHYWJq9whrMda4olJNgk4q9LHt3X9wYLw98N/R6pw=" />
<add key="idpClientSecret" value="pC0NzQQGwyRT5gAsegHbGg==" />
<add key="onbaseRestApiUrl" value="http://
[SampleHostName]/ApiServer/onbase/core/" />
```

3. Ersetzen Sie den Wert des Keys `idpClientId` mit der IDP Client ID.
4. Ersetzen Sie den Wert des Keys `idpUserName` mit dem Benutzernamen eines IDP User Accounts.
5. Ersetzen Sie den Wert des Keys `idpPassword` mit dem verschlüsselten Passwort des IDP User Accounts.
6. Ersetzen Sie den Wert des Keys `idpClientSecret` mit dem verschlüsselten Wert des IDP Secrets.
7. Ersetzen Sie den Hostnamen im Wert des Schlüssels `onbaseRestApiUrl` mit dem Hostnamen des API Servers.
8. Suchen Sie das Element `<configSections>` und stellen Sie sicher, dass die folgende Zeile in diesem Element vorhanden ist.

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

9. Suchen Sie die Zeile, die mit `<odinSettings configSource=` beginnt, und fügen Sie die folgenden Zeilen direkt darunter ein.

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

10. Suchen Sie das Element `<runtime>` und fügen Sie das Element `assemblyBinding` wie in folgendem Beispiel ein.

**Beispiel**

```
<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>
```

11. Speichern und schließen Sie die Datei.
12. Um andere Content Composer-Dienste mit der **OnBase Document API** zu verbinden und die

Monalisa-Bibliothek **MLOnBaseDocumentApi** zur Archivierung von Dokumenten zu verwenden, wiederholen Sie die vorherigen Schritte für die entsprechenden Konfigurationsdateien.

### Anpassen der IdP-Konfigurationsdatei

Die Datei **UserRepository_Idp.config** enthält die allgemeinen IdP-Einstellungen, die von Content Composer Studio und Backend-Anwendungen benötigt werden. Führen Sie die folgenden Schritte auf dem Content Composer Server sowie auf jedem Computer aus, auf dem Content Composer Studio installiert ist.

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **UserRepository_Idp.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Suchen Sie die folgenden Attribute und ändern Sie deren Werte auf die Werte Ihrer IdP-Serverkonfiguration, die von Ihrem IdP-Administrator bereitgestellt werden.
   - `authority`
   - `redirectUri`: Verwenden Sie die redirectUri für Content Composer Studio.
   - `audience`
   - `idpConfigurationURL`

   **Beispiel**

```
<?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. Speichern und schließen Sie die Datei.

## Load Balancer Konfiguration

Um optimale Leistung, Skalierbarkeit und Sicherheit zu erreichen, können Sie die folgenden Content Composer-Anwendungen so konfigurieren, dass sie hinter einem Load Balancer arbeiten:

- Content Composer Server
- Content Composer Studio
- Content Composer Web Client

**Hinweis**: Ein Content Composer-System hinter einem Load Balancer kommuniziert über TCP, daher werden Sticky- oder Cookie-basierte Sitzungen nicht unterstützt.

## Voraussetzungen

- Content Composer Version EP4 oder höher.
- Mindestens zwei Content Composer REST API-Server für die Kommunikation mit Content Composer Studio und Web Client.

  Weitere Informationen finden Sie unter "Information zur Konfiguration des MWS REST API Communication Channels" in der *Content Composer Installation*anleitung
- Ein Front-End-Webserver. Der Content Composer Web Client muss in IIS auf einem anderen System oder einem der Rest-API-Server gehostet werden.
- Ein Load-Balancer-Server und sein DNS-Name.
- Eine Content Composer Studio-Installation auf einem anderen Computer als den Servern.
- Ein Datenbankserver, der von den Content Composer-Servern und dem Content Composer Studio aus zugänglich ist.
- Mindestens zwei verschiedene IP-Ports, einen für die WCF-Kommunikation zwischen den Content Composer-Diensten und Content Composer Studio und einen weiteren Port für die MWS Rest API und Content Composer Web Client.
- Alle Content Composer-Server müssen Zugriff auf das gemeinsame Netzwerkverzeichnis haben, in dem die Konfigurationsdateien gespeichert werden.
- Das Windows-Benutzerkonto, das den Content Composer-Dienst **Composer.Core** ausführt, muss vollen Zugriff auf das gemeinsame Netzwerkverzeichnis haben.
- Ein SSL-Zertificat Weitere Informationen finden Sie unter Installieren des Zertifikats.
- Content Composer-Server und Studio müssen so konfiguriert werden, dass sie alle denselben Port verwenden, der für die externe Netzwerkkommunikation offen ist.
- Content Composer-Server, Studio und Web Client müssen für die Verwendung von HTTPS konfiguriert sein.
- Die Content Composer-Server müssen so konfiguriert werden, dass sie eine gemeinsame Datenbank unter Verwendung des Datenbank-Servernamens verwenden.
- Die Content Composer-Server müssen so konfiguriert sein, dass sie einen gemeinsamen Ordner für den Zugriff auf die gemeinsamen Konfigurationsdateien verwenden.
- Das System muss TLS 1.2 unterstützen.

## Sichern der Konfigurationsdateien

So sichern Sie die Konfigurationsdateien:

- Kopieren Sie auf dem Content Composer-Server aus dem Verzeichnis **%Composerdir%** die folgenden Dateien in ein Sicherungsverzeichnis:
  - Composer.Core.exe.config
  - Composer.MWS.exe.config
  - Composer.OWS.exe.config
  - Composer.XWS.exe.config
  - Composer.Studio.exe.config
  - Composer.WindowsServiceHost.exe.config

## Umgebungsvariablen

Sie können Umgebungsvariablen verwenden, um die Pfade in den Konfigurationsdateien anzugeben.

**Beispiel**

```
<appSettings>
   <add key="rws_config" value="%MySharedPath%\repository.config" />
   <add key="systemlayout_config" value=%MySharedPath%\systemlayout.config"
/>
```

## Installieren des Zertifikats

**Prerequisite**

Ein SSL-Zertifikat von einer Zertifizierungsstelle oder ein selbstsigniertes Zertifikat.

Dieses Zertifikat muss einen Subject Name oder einen alternativen Subject Name haben, der dem DNS-Namen Ihres Load Balancer-Servers und "localhost" für die HTTPS-Kommunikation zwischen den Content Composer-Diensten auf demselben Rechner entspricht.

1. Importieren Sie Ihr SSL-Zertifikat in die Zertifikatspeicher **Personal** und **Trusted Root Certification Authorities** aller Content Composer-Server und Content Composer-Client-Computer.
2. Um das Zertifikat an die Port-Nummer zu binden, führen Sie die folgenden Teilschritte aus:
   1. Öffnen Sie das Zertifikat.
   2. Suchen Sie auf der Registerkarte **Details** das Feld **Thumbprint** und kopieren Sie dessen Wert in die Zwischenablage.
   3. Öffnen Sie ein **Command-Fenster** mit Administratorrechten.
   4. Führen Sie den folgenden Befehl aus und ersetzen Sie dabei `[PortNr]` durch die in den Konfigurationsdateien des Content Composer-Servers verwendete Portnummer und `[Thumbprint]` durch den Thumbprint-Wert Ihres Zertifikats.

      ```
      netsh http add sslcert ipport=0.0.0.0:[PortNr] certhash=
      [Thumbprint] appid=[any GUID]
      ```

**Hinweis**: Für Content Composer Client-Computer ist kein Binding erforderlich.

## IP Network Port Konfiguration

Dieser Abschnitt beschriebt, wie Sie die Konfiguration so ändern, dass für alle Content Composer-Dienste und Content Composer Studio, die WCF-Kommunikation verwenden, nur ein IP-Port verwendet wird. Die Beispiele in diesem Abschnitt verwenden Port 8100.

Folgen Sie den Anweisungen in den folgenden Abschnitten:

- Anpassen der Composer.Core.exe.config zur Verwendung eines IP-Ports
- Anpassen der Composer.MWS.exe.config zur Verwendung eines IP-Ports
- Anpasen der Composer.OWS.exe.config zur Verwendung eines IP-Ports
- Anpassen der Composer.XWS.exe.config zur Verwendung eines IP-Ports
- Ändern der Composer.WindowsServiceHost.exe.config für die Verwendung eines IP-Ports
- Anpassen der Composer.Studio.exe.config zur Verwendung eines IP-Ports

Um Content Composer Web Client zu verwenden, muss der MWS-Rest-API-Endpunkt auf dem Server aktiviert sein, der an einem anderen IP-Port überwacht wird. Weitere Informationen finden Sie unter Content Composer Web Client-Konfiguration für den Load Balancer.

### Anpassen der Composer.Core.exe.config zur Verwendung eines IP-Ports

So ändern Sie die Composer.Core.exe.config:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Core.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Ändern Sie innerhalb des Elements `<appSettings>` alle IP-Ports auf dieselbe Nummer, im folgenden Beispiel `8100`:

   **Beispiel**

   ```
   <appSettings>
       <add key="rws" value="http://localhost:8100/rws" />
       ...
       <add key="sts" value="http://localhost:8100/sts" />
       <add key="login" value="http://localhost:8100/mur/login" />
       <add key="mur" value="http://localhost:8100/mur/data" />
   ```

3. Innerhalb des `<crossDomainServices>` Elements:
   1. Kommentieren Sie die erste Zeile `<add baseAddress` aus.
   2. In der zweiten `<add baseAddress` Zeile:
      - Ändern Sie `https` nach `http`.
      - Ändern Sie `localhost` in den DNS-Namen des Load Balancer-Servers.
      - Ändern Sie den IP-Port nach `8100`.

   **Beispiel**

   ```
   <crossDomainServices policy="%Composerdir%ClientAccessPolicy.xml">
   ```

```
   <baseAddresses>
      <!--     <add baseAddress="http://localhost:8011/"/>     -->
      <add baseAddress="http://[DNS name of the load balancer
server]:8100/"/>
   </baseAddresses>
</crossDomainServices>
```

4. Unterhalb der Zeile `<service name="STS" behaviorConfiguration ="STSBehaviour" >`:
   - Ändern Sie `localhost` in den DNS-Namen des Load Balancer-Servers.
   - Ändern Sie den IP-Port nach `8100`.
   - Fügen sie die Zeile `bindingConfiguration="STS_ WindowsBindingConfiguration"` hinzu.

   **Beispiel**

```
<service name="STS" behaviorConfiguration ="STSBehaviour" >
   <host>
      <baseAddresses>
         <add baseAddress ="http://[DNS name of the load balancer
server]:8100/sts" />
      </baseAddresses>
       </host>
       <endpoint address ="windows"
                 binding="wsHttpBinding"
                 bindingConfiguration="STS_
WindowsBindingConfiguration"
                 contract
="ModusSuite.Runtime.STS.ISecurityTokenService" />
       <endpoint address ="username"
    ....
```

5. Im Element `<add baseAddress` unterhalb der Zeile `<service name="ModusUserRepositoryService" behaviorConfiguration ="ModusSuiteServiceBehaviour" >`:
   - Ändern Sie `localhost` in den DNS-Namen des Load Balancer-Servers.
   - Ändern Sie den IP-Port nach `8100`.

   **Beispiel**

```
<service name="ModusUserRepositoryService" behaviorConfiguration
="ModusSuiteServiceBehaviour">
   <host>
      <baseAddresses>
         <add baseAddress ="http:// [DNS name of the load balancer
server]:8100/mur" />
```

6. Im Element `<add baseAddress` unterhalb der Zeile `<service`

`name="ModusSuite.DataProvider.DataProviderService" behaviorConfiguration ="ModusSuiteServiceBehaviour" >`:

- Ändern Sie `localhost` in den DNS-Namen des Load Balancer-Servers.
- Ändern Sie den IP-Port nach `8100`.

**Beispiel**

```
<service name="ModusSuite.DataProvider.DataProviderService"
behaviorConfiguration ="ModusSuiteServiceBehaviour" >
   <host>
      <baseAddresses>
               <add baseAddress ="http://[DNS name of the load balancer
server]:8100/dataprovider" />
```

7. Im Element `<add baseAddress` unterhalb der Zeile `<service name="ModusSuite.Repository.RepositoryWebService" behaviorConfiguration ="ModusSuiteServiceBehaviour" >`:
   - Ändern Sie `localhost` in den DNS-Namen des Load Balancer-Servers.
   - Ändern Sie den IP-Port nach `8100`.

   **Beispiel**

```
<service name="ModusSuite.Repository.RepositoryWebService"
behaviorConfiguration ="ModusSuiteServiceBehaviour" >
   <host>
      <baseAddresses>
          <add baseAddress ="http:// [DNS name of the load balancer
server]:8100/rws" />
```

8. Speichern und schließen Sie die Datei.

## Anpassen der Composer.MWS.exe.config zur Verwendung eines IP-Ports

So ändern Sie die Composer.MWS.exe.config:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.MWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Ändern Sie innerhalb des Elements `<appSettings>` alle IP-Ports auf dieselbe Nummer, im folgenden Beispiel `8100`:

   **Beispiel**

```
<appSettings>
    ...
    <add key="rws" value="http://localhost:8100/rws" />
    <add key="ows" value="http://localhost:8100/ows/owsrepository" />
    <add key="sts" value="http://localhost:8100/sts" />
    <add key="login" value="http://localhost:8100/mur/login" />
    <add key="mur" value="http://localhost:8100/mur/data" />
    <add key="license" value="http://localhost:8100/mur/license" />
```

```
    ...
</appSettings>
```

3. Unterhalb der Zeile `<service name="MWSRepositoryService" behaviorConfiguration="ModusSuiteServiceBehaviour">`:
   - Ändern Sie `localhost` in den DNS-Namen des Load Balancer-Servers.
   - Ändern Sie den IP-Port nach `8100`.
   - Fügen Sie `mwsrepository` hinzu.
   - Ändern sie `<endpoint address="mwsrepository"` nach `<endpoint address=""`

   **Beispiel**

```
<service name="MWSRepositoryService"
behaviorConfiguration="ModusSuiteServiceBehaviour">
   <host>
      <baseAddresses>
            <add baseAddress=http://[DNS name of the load balancer
server]:8100/mws/mwsrepository />
            </baseAddresses>
   </host>
   <endpoint address="" binding="wsHttpBinding"
bindingConfiguration="TokenBinding" contract="IMWSRepositoryService"
/>
</service>
```

4. Im Element `<add baseAddress` unterhalb der Zeile `service name="ModusSuite.MWS.MWSProcessService" behaviorConfiguration="ModusSuiteServiceBehaviour">`:
   - Ändern Sie `localhost` in den DNS-Namen des Load Balancer-Servers.
   - Ändern Sie den IP-Port nach `8100`.

   **Beispiel**

```
<service name="ModusSuite.MWS.MWSProcessService"
behaviorConfiguration="ModusSuiteServiceBehaviour">
   <host>
      <baseAddresses>
         <add baseAddress="http://[DNS name of the load balancer
server]:8100/mws" />
      </baseAddresses>
   </host>
```

5. Im Element `<add baseAddress` unterhalb der Zeile `<service name="ModusSuite.MWS.MWSProcessServiceBasic" behaviorConfiguration="BasicBehaviour_SSL">`:
   - Ändern Sie `https` nach `http`.

- Ändern Sie `localhost` in den DNS-Namen des Load Balancer-Servers.
- Ändern Sie den IP-Port nach `8100`.

**Beispiel**

```
<service name="ModusSuite.MWS.MWSProcessServiceBasic"
behaviorConfiguration="BasicBehaviour_SSL">
   <host>
      <baseAddresses>
               <add baseAddress="http://[DNS name of the load balancer
server]:8100/mwsbasic" />
      </baseAddresses>
   </host>
   <endpoint address="mwsprocess" binding="basicHttpBinding"
bindingConfiguration="BasicBinding_SSL"
contract="ModusSuite.MWS.Types.IMWSProcessServiceBasic" />
</service>
```

6. Im Element `<add baseAddress` unterhalb der Zeile `<service name="ModusSuite.MWS.MWSProcessServiceRest" behaviorConfiguration="RestBehaviour">`:
   - Ändern Sie `localhost` in den DNS-Namen des Load Balancer-Servers.
   - Ändern Sie den IP-Port nach `8100`.

   **Beispiel**

   ```
   <service name="ModusSuite.MWS.MWSProcessServiceRest"
   behaviorConfiguration="RestBehaviour">
      <host>
         <baseAddresses>
               <add baseAddress="http://[DNS name of the load balancer
   server]:8100/mwsrest" />
         </baseAddresses>
      </host>
   ```

7. Speichern und schließen Sie die Datei.

## Anpasen der Composer.OWS.exe.config zur Verwendung eines IP-Ports

So ändern Sie Composer.OWS.exe.config:

1. Kopieren Sie aus dem Verzeichnis **%Composerdir%** die folgenden Dateien in ein Backup-Verzeichnis. **Hinweis**:
2. Ändern Sie innerhalb des Elements `<appSettings>` alle IP-Ports auf dieselbe Nummer, im folgenden Beispiel `8100`:

   **Beispiel**

   ```
   <appSettings>
      ...
   ```

```
   <add key="rws" value="http://localhost:8100/rws" />
   <add key="sts" value="http://localhost:8100/sts" />
   <add key="login" value="http://localhost:8100/mur/login" />
   <add key="mur" value="http://localhost:8100/mur/data" />
   ...
</appSettings>
```

3. Im Element `<add baseAddress` unterhalb der Zeile `<service name="ModusSuite.OWS.OWSService" behaviorConfiguration="ModusSuiteServiceBehaviour">`.
   - Ändern Sie `localhost` in den DNS-Namen des Load Balancer-Servers.
   - Ändern Sie den IP-Port nach `8100`.

   **Beispiel**

```
<service name="ModusSuite.OWS.OWSService" behaviorConfiguration
="ModusSuiteServiceBehaviour" >
   <host>
      <baseAddresses>
         <add baseAddress ="http://[DNS name of the load balancer
server]:8100/ows" />
      </baseAddresses>
   </host>
   ...
</service>
```

4. Im Element `<add baseAddress` unterhalb der Zeile `<service name="ModusSuite.OWS.OWSServiceBasic" behaviorConfiguration="BasicBehaviour">`.
   - Ändern Sie `localhost` in den DNS-Namen des Load Balancer-Servers.
   - Ändern Sie den IP-Port nach `8100`.

   **Beispiel**

```
<service name="ModusSuite.OWS.OWSServiceBasic" behaviorConfiguration
="BasicBehaviour" >
   <host>
      <baseAddresses>
         <add baseAddress ="http://[DNS name of the load balancer
server]:8100/owsbasic" />
      </baseAddresses>
   </host>
...
</service>
```

5. Speichern und schließen Sie die Datei.

## Anpassen der Composer.XWS.exe.config zur Verwendung eines IP-Ports

So ändern Sie Composer.XWS.exe.config:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.XWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Ändern Sie innerhalb des Elements `<appSettings>` alle IP-Ports auf dieselbe Nummer, im folgenden Beispiel `8100`:

   **Beispiel**

   ```
   <appSettings>
      ...
      <add key="rws" value="http://localhost:8100/rws" />
      <add key="sts" value="http://localhost:8100/sts" />
      <add key="login" value="http://localhost:8100/mur/login" />
      <add key="mur" value="http://localhost:8100/mur/data" />
   ```

3. Im Element `<add baseAddress` unterhalb der Zeile `<service name="ModusSuite.Xdata.SelectionService" behaviorConfiguration="ModusSuiteServiceBehaviour">`:
   - Ändern Sie `localhost` in den DNS-Namen des Load Balancer-Servers.
   - Ändern Sie den IP-Port nach `8100`.

   **Beispiel**

   ```
   <service name="ModusSuite.Xdata.SelectionService"
   behaviorConfiguration ="ModusSuiteServiceBehaviour" >
      <host>
         <baseAddresses>
            <add baseAddress ="http://[DNS name of the load balancer
   server]:8100/xws" />
         </baseAddresses>
      </host>
   ...
   </service>
   ```

4. Im Element `<add baseAddress` unterhalb der Zeile `<service name="ModusSuite.Xdata.SelectionServiceBasic" behaviorConfiguration="BasicBehaviour">`:
   - Ändern Sie `localhost` in den DNS-Namen des Load Balancer-Servers.
   - Ändern Sie den IP-Port nach `8100`.

   **Beispiel**

   ```
   <service name="ModusSuite.Xdata.SelectionServiceBasic"
   behaviorConfiguration ="BasicBehaviour">
      <host>
         <baseAddresses>
   ```

```
            <add baseAddress ="http://[DNS name of the load balancer
server]:8100/xwsbasic" />
         </baseAddresses>
      </host>
...
</service>
```

5. Speichern und schließen Sie die Datei.

## Ändern der Composer.WindowsServiceHost.exe.config für die Verwendung eines IP-Ports

So ändern Sie die Composer.WindowsServiceHost.exe.config:

1. Öffnen Sie im Verzeichnis **%Composerdir%** die Datei **Composer.WindowsServiceHost.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Ändern Sie innerhalb des Elements `<appSettings>` alle IP-Ports auf dieselbe Nummer, im folgenden Beispiel `8100`:

   **Beispiel**

```
<appSettings>
   <add key="rws" value="http://localhost:8100/rws" />
   <add key="ows" value="http://localhost:8100/ows/owsrepository" />
   <add key="sts" value="http://localhost:8100/sts" />
   <add key="login" value="http://localhost:8100/mur/login" />
...
</appSettings>
```

3. Speichern und schließen Sie die Datei.

## Anpassen der Composer.Studio.exe.config zur Verwendung eines IP-Ports

So ändern Sie Composer.Studio.exe.config:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Studio.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Ändern Sie innerhalb des Elements `<appSettings>` alle IP-Ports auf dieselbe Nummer, im folgenden Beispiel `8100`:
   - Ändern Sie alle Vorkommen von `localhost` in den DNS-Namen des Load Balancer-Servers.
   - Ändern Sie alle IP-Ports auf dieselbe Nummer, im folgenden Beispiel `8100`.

   **Beispiel**

```
<appSettings>
   ...
   <add key="sts" value="http://[DNS name of the load balancer
server]:8100/sts" />
   <add key="login" value="http://[DNS name of the load balancer
```

```
server]:8100/mur/login" />
   <add key="license" value="http://[DNS name of the load balancer
server]:8100/mur/license" />
   <add key="rws" value="http://[DNS name of the load balancer
server]:8100/rws" />
   <add key="mwsrws" value="http://[DNS name of the load balancer
server]:8100/mws/mwsrepository" />
   <add key="dpws" value="http://[DNS name of the load balancer
server]:8100/dataprovider" />
   ...
   <add key="ows" value="http://[DNS name of the load balancer
server]:8100/ows/owsrepository" />
```

3. Speichern und schließen Sie die Datei.

## HTTPS Konfiguration für Load Balancer

In diesem Abschnitt werden die Änderungen beschrieben, die in den Konfigurationsdateien für die Verwendung von HTTPS erforderlich sind.

Folgen Sie den Anweisungen in den folgenden Abschnitten:

- Anpassen der Composer.Core.exe.config zur Verwendung von HTTPS
- Anpassen der Composer.MWS.exe.config zur Verwendung von HTTPS
- Anpassen der Composer.OWS.exe.config zur Verwendung von HTTPS
- Anpassen der Composer.XWS.exe.config zur Verwendung von HTTPS
- Anpassen der Composer.WindowsServiceHost.exe.config zur Verwendung von HTTPS
- Anpassen der Composer.Studio.exe.config zur Verwendung von HTTPS

### Anpassen der Composer.Core.exe.config zur Verwendung von HTTPS

So ändern Sie die Composer.Core.exe.config:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Core.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Ändern Sie innerhalb des Elements `<appSettings>` alle `http`-Einträge in `https`.

   **Beispiel**

   ```
   <appSettings>
      <add key="rws" value="https://localhost:8100/rws" />
      ...
      <add key="sts" value="https://localhost:8100/sts" />
      <add key="login" value="https://localhost:8100/mur/login" />
      <add key="mur" value="https://localhost:8100/mur/data" />
   ```

3. Suchen sie nach der Zeile `<service name="remotecontrol" assembly="ModusSuite.Runtime.RemoteControlService" type="ModusSuite.Runtime.RemoteControlService"/>` und kommentieren Sie diese aus.

**Beispiel**

```
<modusruntime>
   <runtimeservices>
   ...
      <!-- <service name="remotecontrol"
assembly="ModusSuite.Runtime.RemoteControlService"
type="ModusSuite.Runtime.RemoteControlService" /> -->
```

4. Suchen Sie die Zeile `<service name="crossdomain" assembly="ModusSuite.Runtime.CrossDomainService" type="ModusSuite.Runtime.CrossDomainRuntimeService"/>` und stellen Sie sicher, dass es nicht auskommentiert ist.

   **Beispiel**

   ```
   <modusruntime>
      <runtimeservices>
      ...
         <service name="crossdomain"
   assembly="ModusSuite.Runtime.CrossDomainService"
   type="ModusSuite.Runtime.CrossDomainRuntimeService"/>
   ```

5. Innerhalb der Elements `<baseAddresses>` unterhalb der Zeile `<crossDomainServices policy="%Composerdir%ClientAccessPolicy.xml">`:
   - Kommentieren Sie das erste Element `<add baseAddress` aus.
   - Im zweiten `<add baseAddress` Element, ändern Sie `localhost` auf den DNS Namen des Load Balancer Servers und den IP Port auf `8100`.

   **Beispiel**

   ```
   <crossDomainServices policy="%Composerdir%ClientAccessPolicy.xml">
      <baseAddresses>
         <!--     <add baseAddress="http://localhost:8011/"/>     -->
         <add baseAddress="https://[DNS name of the load balancer
   server]:8100/"/>
      </baseAddresses>
   </crossDomainServices>
   ```

6. Im Element `<add baseAddress` unterhalb der Zeile `<service name="STS" behaviorConfiguration ="STSBehaviour" >`:
   - Ändern Sie `http` nach `https`.
   - Ändern Sie `localhost` in den DNS-Namen des Load Balancer-Servers.
   - Ändern Sie den IP-Port nach `8100`.
   - Im Element `<endpoint address ="windows"`, fügen Sie `bindingConfiguration="STS_WindowsBindingConfiguration"` hinzu.

   **Beispiel**

   ```
   <service name="STS" behaviorConfiguration ="STSBehaviour" >
   ```

```
   <host>
      <baseAddresses>
         <add baseAddress ="https://[DNS name of the load balancer
server]:8100/sts" />
      </baseAddresses>
   </host>
   <endpoint address ="windows"
             binding="wsHttpBinding"
             bindingConfiguration="STS_WindowsBindingConfiguration"
```

7. Im Element `<add baseAddress` unterhalb der Zeile `<service name="ModusUserRepositoryService" behaviorConfiguration ="ModusSuiteServiceBehaviour" >`:
   - Ändern Sie `http` nach `https`.
   - Ändern Sie `localhost` in den DNS-Namen des Load Balancer-Servers.
   - Ändern Sie den IP-Port nach `8100`.

   **Beispiel**

   ```
   <service name="ModusUserRepositoryService" behaviorConfiguration
   ="ModusSuiteServiceBehaviour" >
      <host>
         <baseAddresses>
            <add baseAddress ="https://[DNS name of the load balancer
   server]:8100/mur" />
         </baseAddresses>
      </host>
   ```

8. Im Element `<add baseAddress` unterhalb der Zeile `<service name="ModusSuite.DataProvider.DataProviderService" behaviorConfiguration ="ModusSuiteServiceBehaviour" >`
   - Ändern Sie `http` nach `https`.
   - Ändern Sie `localhost` in den DNS-Namen des Load Balancer-Servers.
   - Ändern Sie den IP-Port nach `8100`.

   **Beispiel**

   ```
   <service name="ModusSuite.DataProvider.DataProviderService"
   behaviorConfiguration ="ModusSuiteServiceBehaviour" >
      <host>
         <baseAddresses>
            <add baseAddress ="https://[DNS name of the load balancer
   server]:8100/dataprovider" />
         </baseAddresses>
      </host>
   ```

9. Im Element `<add baseAddress` unterhalb der Zeile `<service name="ModusSuite.Repository.RepositoryWebService"`

`behaviorConfiguration ="ModusSuiteServiceBehaviour" >`:

- Ändern Sie `http` nach `https`.
- Ändern Sie `localhost` in den DNS-Namen des Load Balancer-Servers.
- Ändern Sie den IP-Port nach `8100`.

**Beispiel**

```
<service name="ModusSuite.Repository.RepositoryWebService"
behaviorConfiguration ="ModusSuiteServiceBehaviour" >
   <host>
      <baseAddresses>
         <add baseAddress ="https://[DNS name of the load balancer
server]:8100/rws" />
      </baseAddresses>
```

10. Innerhalb des `<bindings>` Elements:
    - Ändern Sie alle Vorkommen von `<security mode="Message">` nach `<security mode="TransportWithMessageCredential">`
    - Fügen Sie ein neues binding namens `STS_WindowsBindingConfiguration` hinzu.

    **Beispiel**

```
<bindings>
   <wsHttpBinding>
      <binding name="TokenBinding" maxBufferPoolSize="2147483600"
maxReceivedMessageSize="2147483600" receiveTimeout="Infinite">
         ...
         <security mode="TransportWithMessageCredential">
         ...
        <binding name="STS_UsernameBindingConfiguration"
maxBufferPoolSize="2147483600" maxReceivedMessageSize="2147483600">
          <security mode="TransportWithMessageCredential">
          ...
        <binding name="STS_WindowsBindingConfiguration"
maxBufferPoolSize="2147483600" maxReceivedMessageSize="2147483600">
          <security mode="TransportWithMessageCredential">
             <message clientCredentialType="Windows" />
          </security>
        </binding>
   </wsHttpBinding>
```

11. Innerhalb des `<behavior name="STSBehaviour">` Elements:
    - Ändern Sie `<serviceMetadata httpGetEnabled="true"/>` nach `<serviceMetadata httpGetEnabled="true" httpsGetEnabled="true" />`.
12. Innerhalb des `<behavior name="ModusSuiteServiceBehaviour">` Elements:
    - Ändern Sie `<serviceMetadata httpGetEnabled="true"/>` nach

```
<serviceMetadata httpGetEnabled="true" httpsGetEnabled="true"
/>.
```

13. Speichern und schließen Sie die Datei.

## Anpassen der Composer.MWS.exe.config zur Verwendung von HTTPS

So ändern Sie die Composer.MWS.exe.config:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.MWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Ändern Sie innerhalb des Elements `<appSettings>` alle `http`-Einträge in `https`.

   **Beispiel**

   ```
   <appSettings>
       ...
       <add key="rws" value="https://localhost:8100/rws" />
       <add key="ows" value="https://localhost:8100/ows/owsrepository" />
       <add key="sts" value="https://localhost:8100/sts" />
       <add key="login" value="https://localhost:8100/mur/login" />
       <add key="mur" value="https://localhost:8100/mur/data" />
       <add key="license" value="https://localhost:8100/mur/license" />
       ...
   </appSettings>
   ```

3. Kommentieren Sie folgenden Zeilen aus:
   - `<service name="debuging" assembly="ModusSuite.Runtime.DebugService" type="ModusSuite.Runtime.DebugService" />`
   - `<service name="remotecontrol" assembly="ModusSuite.Runtime.RemoteControlService" type="ModusSuite.Runtime.RemoteControlService" />`
4. Im Element `<add baseAddress` unterhalb der Zeile `<service name="MWSRepositoryService" behaviorConfiguration="ModusSuiteServiceBehaviour">`, ändern Sie `http` nach `https`.

   **Beispiel**

   ```
   <service name="MWSRepositoryService"
   behaviorConfiguration="ModusSuiteServiceBehaviour">
      <host>
         <baseAddresses>
               <add baseAddress=https://[DNS name of the load balancer
   server]:8100/mws/mwsrepository />
               </baseAddresses>
      </host>
   ```

5. Im Element `<add baseAddress` unterhalb der Zeile `service`

name="ModusSuite.MWS.MWSProcessService" behaviorConfiguration="ModusSuiteServiceBehaviour">, ändern Sie http nach https.

**Beispiel**

```
<service name="ModusSuite.MWS.MWSProcessService"
behaviorConfiguration="ModusSuiteServiceBehaviour">
   <host>
      <baseAddresses>
         <add baseAddress="https://[DNS name of the load balancer
server]:8100/mws" />
      </baseAddresses>
   </host>
```

6. Im Element <add baseAddress unterhalb der Zeile <service name="ModusSuite.MWS.MWSProcessServiceBasic" behaviorConfiguration="BasicBehaviour_SSL">, ändern Sie http nach https.

   **Beispiel**

```
<service name="ModusSuite.MWS.MWSProcessServiceBasic"
behaviorConfiguration="BasicBehaviour_SSL">
   <host>
      <baseAddresses>
               <add baseAddress="https://[DNS name of the load balancer
server]:8100/mwsbasic" />
      </baseAddresses>
   </host>
   <endpoint address="mwsprocess" binding="basicHttpBinding"
bindingConfiguration="BasicBinding_SSL"
contract="ModusSuite.MWS.Types.IMWSProcessServiceBasic" />
</service>
```

7. Im Element <add baseAddress unterhalb der Zeile <service name="ModusSuite.MWS.MWSProcessServiceRest" behaviorConfiguration="RestBehaviour">, ändern Sie http nach https.

   **Beispiel**

```
<service name="ModusSuite.MWS.MWSProcessServiceRest"
behaviorConfiguration="RestBehaviour">
   <host>
      <baseAddresses>
            <add baseAddress="https://[DNS name of the load balancer
server]:8100/mwsrest" />
      </baseAddresses>
   </host>
```

8. Ändern Sie die Zeile `<security mode="Message">` nach `<security mode="TransportWithMessageCredential">`.
9. Im Element `<behavior name="ModusSuiteServiceBehaviour">` ändern Sie `<serviceMetadata httpGetEnabled="true"/>` zu `<serviceMetadata httpGetEnabled="true" httpsGetEnabled="true" />`.
10. Im Element `<behavior name="RestBehaviour">` ändern Sie `<serviceMetadata httpGetEnabled="true"/>` nach `<serviceMetadata httpGetEnabled="true" httpsGetEnabled="true" />`.
11. Speichern und schließen Sie die Datei.

## Anpassen der Composer.OWS.exe.config zur Verwendung von HTTPS

So ändern Sie Composer.OWS.exe.config:

1. Kopieren Sie aus dem Verzeichnis **%Composerdir%** die folgenden Dateien in ein Backup-Verzeichnis. **Hinweis**:
2. Ändern Sie innerhalb des Elements `<appSettings>` alle `http`-Einträge in `https`.

   **Beispiel**

   ```
   <appSettings>
      ...
      <add key="rws" value="https://localhost:8100/rws" />
      <add key="sts" value="https://localhost:8100/sts" />
      <add key="login" value="https://localhost:8100/mur/login" />
      <add key="mur" value="https://localhost:8100/mur/data" />
      ...
   </appSettings>
   ```

3. Kommentieren sie die Zeile `<service name="remotecontrol" assembly="ModusSuite.Runtime.RemoteControlService" type="ModusSuite.Runtime.RemoteControlService" />` aus.

   **Beispiel**

   ```
   <modusruntime enablecache="true">
      <runtimeservices>
         <!-- <service name="remotecontrol"
   assembly="ModusSuite.Runtime.RemoteControlService"
   type="ModusSuite.Runtime.RemoteControlService" /> -->
   ```

4. Im Element `<add baseAddress` unterhalb der Zeile `<service name="ModusSuite.OWS.OWSService" behaviorConfiguration="ModusSuiteServiceBehaviour">`, ändern Sie `http` nach `https`.

   **Beispiel**

   ```
   <service name="ModusSuite.OWS.OWSService"
   behaviorConfiguration="ModusSuiteServiceBehaviour">
   ```

```
   <host>
      <baseAddresses>
            <add baseAddress=https://[DNS name of the load balancer
server]:8100/ows />
```

5. Im Element <add baseAddress unterhalb der Zeile `<service name="ModusSuite.OWS.OWSServiceBasic" behaviorConfiguration="BasicBehaviour">`, ändern Sie `http` nach `https`.

   **Beispiel**

   ```
   <service name="ModusSuite.OWS.OWSServiceBasic" behaviorConfiguration
   ="BasicBehaviour" >
      <host>
         <baseAddresses>
            <add baseAddress ="https://[DNS name of the load balancer
   server]:8100/owsbasic" />
         </baseAddresses>
   ```

6. Ändern Sie die Zeile `<security mode="Message">` nach `<security mode="TransportWithMessageCredential">`.
7. Im Element `<behavior name="ModusSuiteServiceBehaviour">` ändern Sie `<serviceMetadata httpGetEnabled="true"/>` nach `<serviceMetadata httpGetEnabled="true" httpsGetEnabled="true" />`.
8. Speichern und schließen Sie die Datei.

## Anpassen der Composer.XWS.exe.config zur Verwendung von HTTPS

So ändern Sie Composer.XWS.exe.config:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.XWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Ändern Sie innerhalb des Elements `<appSettings>` alle `http`-Einträge in `https`.

   **Beispiel**

   ```
   <appSettings>
      ...
      <add key="rws" value="https://localhost:8100/rws" />
      <add key="sts" value="https://localhost:8100/sts" />
      <add key="login" value="https://localhost:8100/mur/login" />
      <add key="mur" value="https://localhost:8100/mur/data" />
   ```

3. Kommentieren Sie die Zeile `<service name="remotecontrol" assembly="ModusSuite.Runtime.RemoteControlService" type="ModusSuite.Runtime.RemoteControlService" />` aus.

4. Im Element `<add baseAddress` unterhalb der Zeile `<service name="ModusSuite.Xdata.SelectionService" behaviorConfiguration ="ModusSuiteServiceBehaviour" >`, ändern Sie `http` nach `https`.

**Beispiel**

```
<service name="ModusSuite.Xdata.SelectionService"
behaviorConfiguration ="ModusSuiteServiceBehaviour" >
   <host>
      <baseAddresses>
         <add baseAddress ="https://[DNS name of the load balancer
server]:8100/xws" />
      </baseAddresses>
   </host>
...
</service>
```

5. Im Element <add baseAddress unterhalb der Zeile `<service name="ModusSuite.Xdata.SelectionServiceBasic" behaviorConfiguration="BasicBehaviour">`, ändern Sie `http` nach `https`.

**Beispiel**

```
<service name="ModusSuite.Xdata.SelectionServiceBasic"
behaviorConfiguration ="BasicBehaviour">
   <host>
      <baseAddresses>
         <add baseAddress ="https://[DNS name of the load balancer
server]:8100/xwsbasic" />
      </baseAddresses>
   </host>
...
</service>
```

6. Ändern Sie die Zeile `<security mode="Message">` zu `<security mode="TransportWithMessageCredential">`.
7. Im Element `<behavior name="ModusSuiteServiceBehaviour">` ändern Sie `<serviceMetadata httpGetEnabled="true"/>` nach `<serviceMetadata httpGetEnabled="true" httpsGetEnabled="true" />`.
8. Speichern und schließen Sie die Datei.

## Anpassen der Composer.WindowsServiceHost.exe.config zur Verwendung von HTTPS

So ändern Sie die Composer.WindowsServiceHost.exe.config:

1. Öffnen Sie im Verzeichnis **%Composerdir%** die Datei **Composer.WindowsServiceHost.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Ändern Sie innerhalb des Elements `<appSettings>` alle `http`-Einträge in `https`.

**Beispiel**

```
<appSettings>
   <add key="rws" value="https://localhost:8100/rws" />
   <add key="ows" value="https://localhost:8100/ows/owsrepository" />
```

```
   <add key="sts" value="https://localhost:8100/sts" />
   <add key="login" value="https://localhost:8100/mur/login" />
...
</appSettings>
```

3. Fügen Sie innerhalb des Elements <bindings> ein neues Element <binding name> hinzu, wie im folgenden Beispiel gezeigt:

   **Beispiel**

```
<wsHttpBinding>
   <binding name="TokenBinding" maxBufferPoolSize="2147483600"
maxReceivedMessageSize="2147483600" receiveTimeout="Infinite">
      <readerQuotas maxStringContentLength="2147483647"
maxArrayLength="2147483647" />
      <security mode="TransportWithMessageCredential">
         <message clientCredentialType="IssuedToken"
establishSecurityContext="false"/>
      </security>
   </binding>
   ...
```

4. Speichern und schließen Sie die Datei.

## Anpassen der Composer.Studio.exe.config zur Verwendung von HTTPS

So ändern Sie Composer.Studio.exe.config:

1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Studio.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
2. Ändern Sie innerhalb des Elements <appSettings> alle http-Einträge in https. **Beispiel**

```
<appSettings>


   ...
   <add key="sts" value="https://[DNS name of the load balancer
server]:8100/sts" />
   <add key="login" value="https://[DNS name of the load balancer
server]:8100/mur/login" />
   <add key="license" value="https://[DNS name of the load balancer
server]:8100/mur/license" />
   <add key="rws" value="https://[DNS name of the load balancer
server]:8100/rws" />
   <add key="mwsrws" value="https://[DNS name of the load balancer
server]:8100/mws/mwsrepository" />
   <add key="dpws" value="https://[DNS name of the load balancer
server]:8100/dataprovider" />
   ...
```

```
   <add key="ows" value="https://[DNS name of the load balancer
server]:8100/ows/owsrepository" />
```

3. Speichern und schließen Sie die Datei.

## Geteilte Konfigurationsdateien

In diesem Abschnitt beschreibt die erforderlichen Änderungen, damit zwei oder mehr Content Composer-Installationen geteilte Konfigurationsdateien verwenden können.

### Einrichten der ersten Umgebung

Um die erste Content Composer-Installation in Ihrer Umgebung einzurichten, führen Sie die folgenden Schritte aus.

1. Führen Sie die Schritte unter "Konfigurieren des Content Composer-Servers" in der *Content Composer-Installationsanleitung* unter Verwendung der IP-Adresse oder des DNS-Namens des geteilten Datenbankservers aus.
2. Befolgen Sie die Anweisungen zur Verschlüsselung, die unter "Informationen zur RSA-Verschlüsselung" in der *Content Composer-Installationsanleitung* beschrieben sind.
3. Erstellen Sie auf einer Netzwerkfreigabe ein neues geteiltes Verzeichnis.
4. Kopieren Sie die folgenden Dateien aus dem Verzeichnis **%Composerdir%** in das neu erstellte geteilte Verzeichnis.
   - repository.config
   - systemlayout.config
   - UserRepository.config
   - UserRepository_Ldap.config
   - UserRepository_Idp.config
   - %ComposerDir%\Admin\rws_work
5. So ändern Sie die **Composer.Core.exe.config**:
   1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Core.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
   2. Im Element `<appSettings>` ändern Sie die Werte der Keys `rws_config` und `systemlayout_config` auf das zuvor erstellte geteilte Verzeichnis.

      **Beispiel**

      ```
      <appSettings>
         ...
         <add key="rws_config"
      value="
      \\SharedFolderDnsName\SharedConfigDir\repository.config" />
         <add key="systemlayout_config"
      value="
      \\SharedFolderDnsName\SharedConfigDir\systemlayout.config" />
      ```

   3. Im Element `<appSettings>` unterhalb des Keys `EnableHylandLogging` fügen Sie

die neuen Keys wie in folgendem Beispiel ein und setzen Sie die Werte der Keys `userRepository`, `userRepository_Ldap`, and `userRepository_Idp` auf das zuvor erstellte geteilte Verzeichnis.

**Beispiel**

```
<appSettings>
   ...
   <add key="userRepository"
value="\\SharedFolderDnsName\SharedConfigDir\UserRepository.co
nfig" />
   <add key="userRepository_Ldap"
value="\\SharedFolderDnsName\SharedConfigDir\UserRepository_
ldap.config" />
   <add key="userRepository_Idp" value="\\
SharedFolderDnsName\SharedConfigDir\UserRepository_Idp.config"
/>
```

4. Erstellen Sie auf allen Content Composer-Serverinstallationen eine neue Umgebungsvariable namens `sharedDir` und setzen Sie den Wert auf das zuvor erstellte geteilte Verzeichnis.

   **Beispiel**

   ```
   sharedDir=\\SharedFolderDnsName\SharedConfigDir
   ```

5. Im Element `<appSettings>` setzen Sie den Wert des Keys `rws_work_dir` wie in folgendem Beispiel gezeigt:

   **Beispiel**

   ```
   <appSettings>
      ...
      <add key="rws_work_dir" value="%sharedDir%\Admin\rws_work"
   />
   ```

6. Speichern und schließen Sie die Datei.

6. So ändern Sie die **Composer.MWS.exe.config**:
   1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.MWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
   2. Im Element `<appSettings>` unterhalb des Keys `EnableHylandLogging` fügen Sie die neuen Keys wie in folgendem Beispiel ein und setzen Sie die Werte der Keys `userRepository`, `userRepository_Ldap`, and `userRepository_Idp` auf das zuvor erstellte geteilte Verzeichnis.

      **Beispiel**

      ```
      <appSettings>
         ...
         <add key="userRepository"
      ```

```
value="\\SharedFolderDnsName\SharedConfigDir\UserRepository.co
nfig" />
   <add key="userRepository_Ldap"
value="\\SharedFolderDnsName\SharedConfigDir\UserRepository_
ldap.config" />
   <add key="userRepository_Idp" value="\\
SharedFolderDnsName\SharedConfigDir\UserRepository_Idp.config"
/>
```

3. Speichern und schließen Sie die Datei.

7. Stellen Sie sicher, dass das Windows-Benutzerkonto, das für die Ausführung der Dienste **Composer.Core** und **Composer.MWS** konfiguriert ist, volle Zugriffsrechte auf das zuvor erstellte geteilte Verzeichnis hat.
8. Um die Konfigurationsänderungen zu aktivieren, starten Sie in den Windows-Diensten die Dienste **Composer.Core** und **Composer.MWS** neu.

## Einrichten einer zusätzlichen Umgebung

Um eine zusätzlcihe Content Composer-Installation in Ihrer Umgebung einzurichten, führen Sie die folgenden Schritte aus.

1. Führen Sie eine Installation von Content Composer durch, die mindestens das Feature Content Composer Server enthält. Weitere Informationen finden Sie unter "Durchführen einer vollständigen Installation von Content Composer" in der *Content Composer-Installationsanleitung*.

   **Wichtig**: Führen Sie **Composer.SetupAssistantnicht** aus, wenn Sie einen zusätzlichen Composer-Server installieren.
2. Befolgen Sie die Anweisungen zur Verschlüsselung, die unter "Informationen zur RSA-Verschlüsselung" in der *Content Composer-Installationsanleitung* beschrieben sind.
3. Kopieren Sie die folgenden Dateien aus der Erstinstallation in das Verzeichnis **% Composerdir%**der zusätzlichen Serverinstallation:
   - Composer.lic
   - Alle vorhandenen DOTM-Dateien
   - Alle CONFIG-Dateien
4. Erstellen Sie auf allen Content Composer-Serverinstallationen eine neue Umgebungsvariable namens `sharedDir` und setzen Sie den Wert auf das geteilte Verzeichnis.

   **Beispiel**

   `sharedDir=\\SharedFolderDnsName\SharedConfigDir`
5. So ändern Sie die **Composer.Core.exe.config**:
   1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Core.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
   2. Im Element `<appSettings>` ändern Sie die Werte der Keys `rws_config` und `systemlayout_config` auf das geteilte Verzeichnis.

      **Beispiel**

```
<appSettings>
   ...
   <add key="rws_config"
value="
\\SharedFolderDnsName\SharedConfigDir\repository.config" />
   <add key="systemlayout_config"
value="
\\SharedFolderDnsName\SharedConfigDir\systemlayout.config" />
```

3. Im Element `<appSettings>` unterhalb des Keys `EnableHylandLogging` fügen Sie die neuen Keys wie in folgendem Beispiel ein und setzen Sie die Werte der Keys `userRepository`, `userRepository_Ldap`, and `userRepository_Idp` auf das zuvor erstellte geteilte Verzeichnis.

   **Beispiel**

```
<appSettings>
   ...
   <add key="EnableHylandLogging" value="False" />
   <add key="userRepository"
value="\\SharedFolderDnsName\SharedConfigDir\UserRepository.co
nfig" />
   <add key="userRepository_Ldap"
value="\\SharedFolderDnsName\SharedConfigDir\UserRepository_
ldap.config" />
   <add key="userRepository_Idp" value="\\
SharedFolderDnsName\SharedConfigDir\UserRepository_Idp.config"
/>
   <add key="EnableSecureContextMessageHandshake"
value="false"/>
```

4. Im Element `<appSettings>` setzen Sie den Wert des Keys `rws_work_dir` wie in folgendem Beispiel

   gezeigt: **Beispiel**

```
<appSettings>
   ...
   <add key="rws_work_dir" value="%sharedDir%\Admin\rws_work"
/>
```

5. Speichern und schließen Sie die Datei.

6. So ändern Sie die **Composer.MWS.exe.config**:
   1. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.MWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
   2. Suchen Sie nach der Zeile, die mit `<mws systemoid=`.
   3. Ersetzen Sie die Zeile durch die entsprechende Zeile aus der **Composer.MWS.exe.config** in der Ausgangsumgebung.

**Beispiel**

```
<mws systemoid="TestSystemOID" />
```

4. Speichern und schließen Sie die Datei.

7. Um die Konfigurationsänderungen zu aktivieren, starten Sie in den Windows-Diensten die Dienste **Composer.Core** und **Composer.MWS** neu.

## Verwenden des Studio über den Load Balancer Server

Dieser Abschnitt beschreibt die Änderungen, die erforderlich sind, damit Content Composer Studio mit Content Composer Server über den Load Balancer kommunizieren kann.

**Prerequisite** Die Schritte in den Abschnitten

- IP Network Port Konfiguration
- HTTPS Konfiguration für Load Balancer
- Einrichten der ersten Umgebung
- Einrichten einer zusätzlichen Umgebung

wurden ausgeführt.

1. Öffnen Sie auf dem Server, auf dem Content Composer Studio installiert ist, im Verzeichnis **%Composerdir%** die Datei **Composer.Core.exe.config** mit einem Texteditor, der UTF-8 unterstützt, und nehmen Sie die folgenden Änderungen vor:
    1. Suchen Sie das Element `<wsHttpBinding>` und ersetzen Sie das Element mit dem folgenden:

```
<wsHttpBinding>
   <binding name="TokenBinding"
maxBufferPoolSize="2147483600"
maxReceivedMessageSize="2147483600"
receiveTimeout="Infinite">
      <readerQuotas maxStringContentLength="2147483647"
maxArrayLength="2147483647" />
      <security mode="TransportWithMessageCredential">
         <message clientCredentialType="IssuedToken"
establishSecurityContext="false" />
      </security>
   </binding>
   <!-- Settings for the communication with the repository
(rws)  DON'T DELETE the binding element, because the element
will read out at the creation of the channel  -->
   <binding name="rws" sendTimeout="00:05:00" />
   <binding name="STS_UsernameBindingConfiguration"
maxBufferPoolSize="2147483600"
maxReceivedMessageSize="2147483600">
      <security mode="TransportWithMessageCredential">
         <message clientCredentialType="UserName"
```

```
establishSecurityContext="false" />
      </security>
   </binding>
   <binding name="STS_WindowsBindingConfiguration"
maxBufferPoolSize="2147483600"
maxReceivedMessageSize="2147483600">
      <security mode="TransportWithMessageCredential">
         <message clientCredentialType="Windows"
establishSecurityContext="false" />
      </security>
   </binding>
</wsHttpBinding>
```

2. Fügen Sie im Element `<appSettings>` die folgende Zeile hinzu:

```
<add key="EnableSecureContextMessageHandshake"
value="false"/>
```

3. Speichern und schließen Sie die Datei.

2. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.MWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt und nehmen Sie folgende Änderung vor:
   1. Suchen Sie das Element `<binding name="TokenBinding"` und ersetzen Sie das Element mit dem folgenden:

```
<binding name="TokenBinding" maxBufferPoolSize="2147483600"
maxReceivedMessageSize="2147483600"
receiveTimeout="Infinite">
   <readerQuotas maxStringContentLength="2147483647"
maxArrayLength="2147483647" />
   <security mode="TransportWithMessageCredential">
      <message clientCredentialType="IssuedToken"
establishSecurityContext="false"/>
   </security>
</binding>
```

   2. Fügen Sie im Element `<appSettings>` die folgende Zeile hinzu:

```
<add key="EnableSecureContextMessageHandshake"
value="false"/>
```

   3. Speichern und schließen Sie die Datei.
3. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.OWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
   1. Suchen Sie das Element `<binding name="TokenBinding"` und ersetzen Sie das Element mit dem folgenden:

```
<binding name=”TokenBinding” maxBufferPoolSize=”2147483600”
maxReceivedMessageSize=”2147483600”
```

```
receiveTimeout=”Infinite”>
   <readerQuotas maxStringContentLength=”2147483647”
maxArrayLength=”2147483647” />
   <security mode=”TransportWithMessageCredential”>
      <message clientCredentialType=”IssuedToken”
establishSecurityContext=”false” />
   </security>
</binding>
```

2. Fügen Sie im Element `<appSettings>` die folgende Zeile hinzu:

```
<add key=”EnableSecureContextMessageHandshake”
value=”false”/>
```

3. Speichern und schließen Sie die Datei.

4. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.XWS.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
    1. Suchen Sie das Element `<binding name="TokenBinding"` und ersetzen Sie das Element mit dem folgenden:

```
<binding name=”TokenBinding” maxBufferPoolSize=”2147483600”
maxReceivedMessageSize=”2147483600”
receiveTimeout=”Infinite”>
   <readerQuotas maxStringContentLength=”2147483647”
maxArrayLength=”2147483647” />
   <security mode=”TransportWithMessageCredential”>
      <message clientCredentialType=”IssuedToken”
establishSecurityContext=”false” />
   </security>
</binding>
```

    2. Fügen Sie im Element `<appSettings>` die folgende Zeile hinzu:

```
<add key=”EnableSecureContextMessageHandshake”
value=”false”/>
```

    3. Speichern und schließen Sie die Datei.

5. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.Studio.exe.config** mit einem Texteditor, der UTF-8 unterstützt.
    1. Suchen Sie das Element `<wsHttpBinding>` und ersetzen Sie das Element mit dem folgenden:

```
<wsHttpBinding>
   <!-- Settings for the communication with the repository
(rws) DON'T DELETE the binding element, because the element
will read out at the creation of the channel  -->
   <binding name="TokenBinding"
maxBufferPoolSize="2147483600"
```

```
maxReceivedMessageSize="2147483600"
receiveTimeout="Infinite">
      <readerQuotas maxStringContentLength="2147483647"
maxArrayLength="2147483647" />
      <security mode="TransportWithMessageCredential">
         <message clientCredentialType="IssuedToken"
establishSecurityContext="false"/>
      </security>
   </binding>
   <binding name="rws" sendTimeout="00:05:00" />
   <binding name="mws" receiveTimeout="00:05:00"
sendTimeout="00:05:00" />
   <!-- configuration for monalisa extension 'MLMwsClient'-->
</wsHttpBinding>
```

2. Innerhalb des Elements `<appSettings>` ändern Sie alle Vorkommen von `http` nach `https`, alle Vorkommen von `localhost` in den DNS-Namen des Load Balancer Servers, und alle IP-Ports in diesselbe Nummer, im folgenden Beispiel `8100`: **Beispiel**

```
<appSettings>
   ...
   <add key="sts" value="https://[DNS name of the load
balancer server]:8100/sts" />
   <add key="login" value="https://[DNS name of the load
balancer server]:8100/mur/login" />
   <add key="license" value="https://[DNS name of the load
balancer server]:8100/mur/license" />
   <add key="rws" value="https://[DNS name of the load
balancer server]:8100/rws" />
   <add key="mwsrws" value="https://[DNS name of the load
balancer server]:8100/mws/mwsrepository" />
   <add key="dpws" value="https://[DNS name of the load
balancer server]:8100/dataprovider" />
   ...
   <add key="ows"
value="
https://LoadBalancerServerExample:8100/ows/owsrepository" />
```

3. Fügen Sie im Element `<appSettings>` die folgende Zeile hinzu:

```
<add key="EnableSecureContextMessageHandshake"
value="false"/>
```

4. Speichern und schließen Sie die Datei.

6. Öffnen Sie aus dem Verzeichnis **%Composerdir%** die Datei **Composer.WindowsServiceHost.exe.config** mit einem Texteditor, der UTF-8 unterstützt.

1. Fügen Sie im Element `<appSettings>` die folgende Zeile hinzu:

```
<add key="EnableSecureContextMessageHandshake"
value="false"/>
```

2. Suchen Sie das Element `<wsHttpBinding>` und ersetzen Sie das Element mit dem folgenden:

```
<wsHttpBinding>
   <binding name="TokenBinding"
maxBufferPoolSize="2147483600"
maxReceivedMessageSize="2147483600"
receiveTimeout="Infinite">
      <readerQuotas maxStringContentLength="2147483647"
maxArrayLength="2147483647" />
      <security mode="TransportWithMessageCredential">
         <message clientCredentialType="IssuedToken"
establishSecurityContext="false"/>
      </security>
   </binding>
   <!-- Settings for the communication with the repository
(rws)
   DON'T DELETE the binding element, because the element will
read out at the creation of the channel  -->
   <binding name="rws" sendTimeout="00:05:00"/>
   <binding name="mws" receiveTimeout="00:05:00"
sendTimeout="00:05:00" /> <!-- configuration for monalisa
extension 'MLMwsClient'-->
</wsHttpBinding>
```

3. Speichern und schließen Sie die Datei.

## Content Composer Web Client-Konfiguration für den Load Balancer

Dieser Abschnitt beschreibt die Änderungen, die erforderlich sind, damit Content Composer Web Client mit Content Composer Server über den Load Balancer kommunizieren kann.

### Vorbereiten des Content Composer Rest API Servers

**Voraussetzung** Die Schritte in den Abschnitten

- IP Network Port Konfiguration
- HTTPS Konfiguration für Load Balancer
- Einrichten der ersten Umgebung
- Einrichten einer zusätzlichen Umgebung
- Verwenden des Studio über den Load Balancer Server
- " Information zur Konfiguration des MWS REST API Communication Channels" in der *Content Composer Installation*anleitung

wurden ausgeführt.

Führen Sie die folgenden Schritte aus:

1. Starten Sie jedes Content Composer Studio, das hinter einem Load Balancer installiert ist.
2. Wechseln Sie in das Standardsystem.
3. Öffnen Sie im **Navigator** den Prozess **MWS_Standard**.
4. Ändern Sie im **Objektinspektor** die Eigenschaft **Prozessverzeichnis** in ein geteiltes, für alle Server zugängliches Verzeichnis.
5. Klicken Sie auf der Registerkarte **Prozess Designer Tools** auf **Speichern**.
6. Starten Sie das Content Composer Studio.

### Installation des Content Composer Web Clients

1. Installieren und konfigurieren Sie den Content Composer Web Client. Weitere Informationen finden Sie in der Content Composer Web Client-Installationsanleitung.
2. Öffnen Sie die Datei **init-config.json** im Verzeichnis *[Installationspfad]\hy-app\configs* in einem Texteditor, der UTF-8 unterstützt.
3. Suchen Sie die folgende Zeile und erstzen Sie `<DNS name of the load balancer server>` mit der IP oder dem DNS-Namen Ihres **Load Balancer Servers**.

   ```
   "baseApiUrl": https://<DNS name of the load balancer server>:9010/mws
   ```

4. Speichern und schließen Sie die Datei.
5. Stellen Sie sicher, dass Ihre Firewall die Kommunikation über Port **9010** zulässt.

## Beispiel für eine Logging-Konfiguration

### Beispiel einer Frontend-Load-Balancer-Konfiguration für Content Composer Web Client

```
frontend ccm_clients_web
bind *:9010
mode tcp
option tcplog
default_backend ccm_servers_web
...
```

### Beispiel einer Backend-Load-Balancer-Konfiguration für Content Composer Web Client

```
backend ccm_servers_web
mode tcp
option tcplog
balance roundrobin
server cocoserver1:9010 ...
```

```
server cocoserver2:9010 ...
...
```

### Beispiel für eine Frontend-Load-Balancer-Konfiguration für Content Composer Studio

```
frontend ccm_clients_studio
bind *:8100
mode tcp
option tcplog
default_backend ccm_servers_studio
...
```

### Beispiel einer Backend-Load-Balancer-Konfiguration für Content Composer Studio

```
backend ccm_servers_studio
mode tcp
option tcplog
balance roundrobin
server cocoserver1:8100 ...
server cocoserver2:8100 ...
...
```