Sitecore Experience Platform Upgrade Guide

Sitecore Experience Platform 9.2.0

Upgrade to Sitecore XP 9.2.0 from Sitecore XP 9.0.2 or later

Table of Contents

Chapter 1

Introduction

This guide describes how to upgrade from Sitecore Experience Platform 9.0.2 or later to Sitecore Experience Platform 9.2.0 and applies only to the on-premise version.

If you are upgrading from Sitecore XP 9.0.1 or earlier, use the *Upgrade to Sitecore XP 9.2.0 from Sitecore XP 8.1.0 - 9.0.1 guide* available at the Sitecore Downloads site.

The tasks and processes in this guide require local administrator permissions.

Before upgrading your Sitecore XP production system, you should first test the upgrade process on a test or developer system. Testing the upgrade process can highlight issues with custom tools or applications.

Sitecore XP uses an in-place upgrade process. Once you are ready to apply upgrade to your production system, inform your Sitecore users that all of the Sitecore functionality will not be available during the process.

1.1 Prerequisites

Before you upgrade to Sitcore XP 9.2.0, ensure that you meet the software and hardware requirements for this version of Sitecore XP.

For more information about the requirements, see the Sitecore XP 9.2.0 Installation Guide.

Chapter 2

Preparing to Upgrade Sitecore XP

Before upgrading, you must download the upgrade files and prepare your environment for the upgrade process.

This chapter contains the following sections:

- Files and scripts for upgrading
- Preupgrade steps for Sitecore XP

2.1 Files and scripts for upgrading

This section lists the files required to upgrade Sitecore XP. Download the files from the **Upgrade options** section on the Sitecore Experience Platform download site.

To upgrade from Sitecore XP 9.0.2, you require the following file:

- `Sitecore Update Installation Wizard 4.0.1 rev. 00153 for 8.x-9.0.zip`

To upgrade from Sitecore XP 9.1.0, you require the following file:

- `Sitecore Update Installation Wizard 4.0.1 rev. 00153 for 9.1.zip`

To upgrade from Sitecore XP 9.0.2, 9.1.0, and 9.1.1, you also require the following file which is noted as **Configuration files for upgrade** on the download site:

- `Sitecore 9.2.0. rev. 002893 (upgrade files).zip`

  This file contains:

  - `Database Upgrade Script.zip`

    This file contains a number of files, including the following database upgrade scripts:

    - `CMS_core.sql`
    - `SXP_reporting.sql`
    - `SXP_reporting_migrate_experience_analytics_data.sql`
    - `SXP_processing_pools.sql`
    - `SXP_messaging.sql`
    - `SXP_exm_master_dispatchqueue.sql`
    - `SXP_exm_master_save_suppressions.sql`
    - `SXP_exm_master_suppressions.sql`
    - `SXP_exm_master_update_suppressions.sql`

    The upgrade scripts contain changes that must be applied to Sitecore databases across Sitecore XP to ensure compatibility. They also modify certain tables and collections to support new functionality and improve performance. The changes must be applied once per database.

2.2 Preupgrade steps for Sitecore XP

This section describes the steps that you must perform before starting the upgrade process.

2.2.1 Stop active content tests

Content tests normally stop automatically after enough data has been gathered for a winner to be picked. However, before upgrading your Sitecore installation, you must stop the active content tests by either having the tests select a winner or by manually canceling the tests.

2.2.2 Upgrade the databases

The upgrade scripts contain changes that must be applied to Sitecore databases across Sitecore XP to ensure compatibility. They also modify certain tables and collections to support new functionality and improve performance. The changes must be applied once per database.

These scripts are available in the `Database Upgrade Script.zip` file.

To upgrade the databases:

1. Run the `CMS_core.sql` script for the *Core* database..
2. Run the `SXP_reporting.sql` script for the *Reporting* database.
3. If you are for upgrading from Sitecore XP 9.0.2, run the `SXP_exm_master_dispatchqueue.sql` script for the *EXM.Master* database.
4. Run the `SXP_exm_master_suppressions.sql` script for the *EXM.Master* database.
5. Run the `SXP_exm_master_save_suppressions.sql` script for the *EXM.Master* database.
6. Run the `SXP_exm_master_update_suppressions.sql` script for the *EXM.Master* database.

2.2.3 Disable Sitecore modules

You must disable any Sitecore modules that are in use.

If you are using Sitecore Experience Accelerator (SXA) and want to upgrade the platform, follow the instructions in the SXA Upgrade Guide. The key step to disable SXA is to move DLL files to a temporary location.

2.2.4 Disable Web Forms for Marketers

The Web Forms for Marketers module was deprecated with the release of Sitecore XP 9.1. If you are upgrading from Sitecore XP 9.0.2 and are using the Web Forms for Marketers module, you must disable it before upgrading.

To disable the Web Forms for Marketers module:

- Disable the following Web Forms for Marketers configuration files by adding `.disabled` to the file extension.
    - `\App_Config\Include\Sitecore.WFFM.Speak.config`
    - `\App_Config\Include\Sitecore.WFFM.Services.config`

- \App_Config\Include\Sitecore.WFFM.Dependencies.config
- \App_Config\Include\Sitecore.WFFM.Analytics.config
- \App_Config\Include\Sitecore.Forms.config
- \App_Config\Include\Sitecore.MvcForms.config

2.2.5 Disable active Sitecore services

Before upgrading, you must disable some active services.

To disable the active services:

1. Delete all the `*.JavaScriptServices.*` files from the `bin` and `App_config` folders - and any child folders.

   If you are upgrading from Sitecore XP 9.0.2, these files are in the root `Website` folder.
2. Remove the `Sitecore.LayoutService.Jss.config` file from the `App_Config\Sitecore\LayoutService\` folder.
3. Remove the `zSitecore.JavaScriptServices.GraphQL.config` file from `App_Config\Sitecore\Services.GraphQL\` folder.

2.2.6 Disable custom DLLs

The new version of Sitecore might contain some breaking changes. To separate the upgrade of Sitecore from the upgrade of any customizations you have made, move your custom DLL and configuration files out of `sitecore \bin` and `\app_config` folders. When the upgrade is completed, you can adapt your code according to a list of breaking changes and move the files back.

2.2.7 Add the Security Connection String

Important
This section applies only if you're upgrading from Sitecore 9.0.2.

Since Sitecore XP 9.1.0, you can move security from the *Core* database to another database.

Before you upgrade to Sitecore XP 9.2.0, you must add the *security* connection string.

To add the *security* connection string, in the `\App_Config` folder, in the `ConnectionStrings.config` file, add for example:

```
<add name="security" connectionString="user id=user;password=password;Data
Source=(server);Database=Sitecore.Core" />
```

Chapter 3

Using the Sitecore Update Center

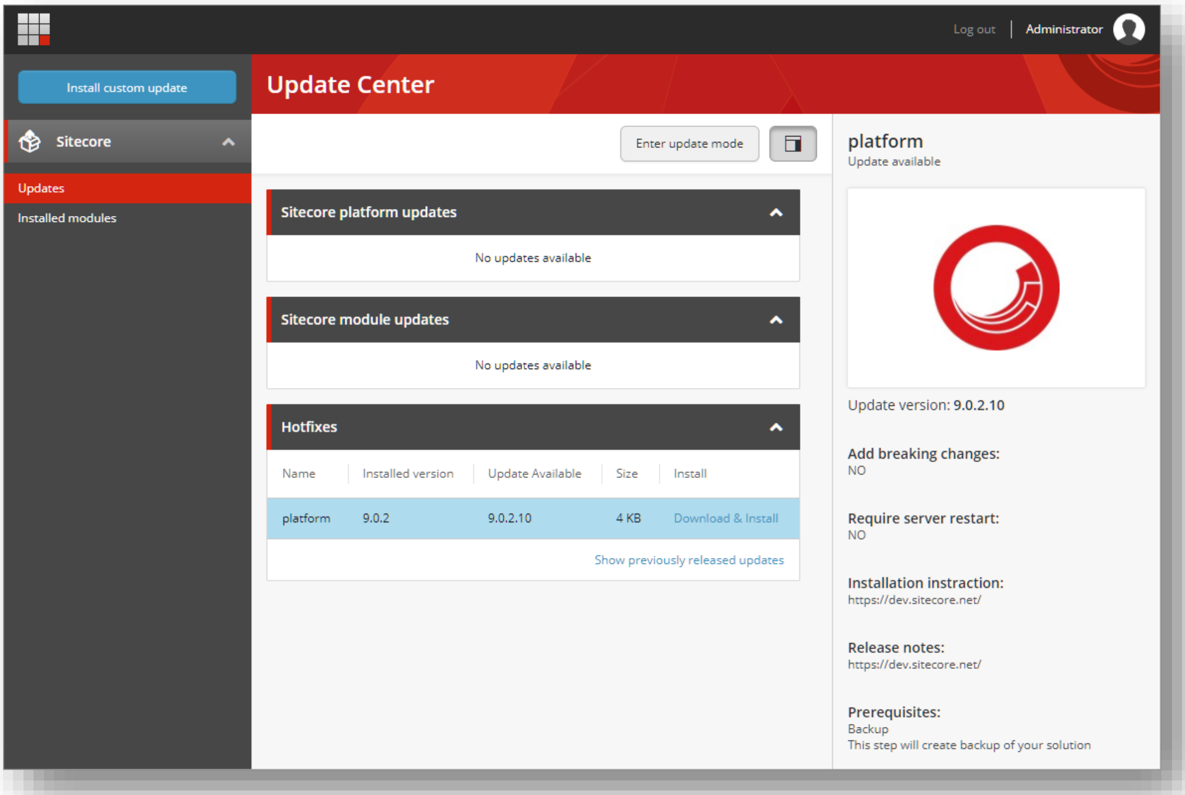
You can use the Sitecore Update Center to find, download, install, and manage upgrades and hotfixes for the Sitecore platform and modules. This tool was introduced in Sitecore XP 9.0.2.

This chapter contains the following sections:

3.1 Sitecore Update Center

You use the Update Center to find, download, install, and manage upgrades and hotfixes for the Sitecore platform and modules. The Update Center accesses a package management service for information about upgrade and for the upgrade packages themselves. Sitecore provides a package management service, or you can install and provide a service yourself. Finally, the Update Center can install updates from a package in the file system.

You open Update Center from the Sitecore Dashboard:



Sitecore platform updates

No updates available

Sitecore module updates

No updates available

Hotfixes

Name	Installed version	Update Available	Size	Install
platform	9.0.2	9.0.2.10	4 KB	Download & Install

Show previously released updates

platform
Update available

Update version: 9.0.2.10

Add breaking changes:
NO

Require server restart:
NO

Installation instraction:
https://dev.sitecore.net/

Release notes:
https://dev.sitecore.net/

Prerequisites:
Backup
This step will create backup of your solution

The left pane, contains three options:

- **Updates** – displays a list of the available upgrades and updates from the package management service.
- **Installed modules** – displays a list of modules that are already installed.
- **Install custom update** – click this option to select an update package in the file system.

When you select an update package from the list of available updates, the Update Center displays information about the selected package. This information is from the package meta data, and it typically includes:

- Version number.
- Information about breaking changes.
- A link to the release note for the packages.
- Pre- and post-installation steps.

The final element of the Update Center is the **Enter update mode/Exit update mode** button that toggles the update mode on and off.

When you install an update, you use the **Ready to install** window to set the options and start the process.

3.1.1 Update mode

When you install upgrades, we recommended that you enter Update mode.

The Sitecore configuration is divided into four layers:

- Sitecore
- Modules
- Custom
- Environment

When Sitecore is in Update mode, it turns off all the configuration layers except the Sitecore layer. Additionally, it adds and activates a layer called `Update`, and creates a temporary folder called `Update in the App_Config` folder.

The `Update` folder contains these configuration files:

- `DisableXDB.config` – this file disables Sitecore xDB and Sitecore xDB Tracking.
- `DataFolder.config` – this file enables the Data folder. This is necessary because Sitecore does not work without access to the license.xml file.
- `Layers.config.orig` – this file is a copy of how the Layers.config file was when Sitecore entered update mode.

When Sitecore is in update mode, it runs as platform-only. This provides more stability while the update takes place. When Sitecore exits update mode, it restores the `Layers.config` file and deletes the temporary `Update` folder.

3.1.2 Configuration

You specify the package management service the Update Center connects to in the `App_Config/ConnectionStrings.config` file.

The setting is `PackageManagementServiceUrl`. The default setting connects to Sitecore package management:

```
<add name="PackageManagementServiceUrl"
connectionString="https://updatecenter.cloud.sitecore.net/" />
```

You can point to a local instance of the package management service, if you have set up such a service. For example:

```
<add name="PackageManagementServiceUrl"
connectionString="https://someserver.local/packagemanagementservice/" />
```

3.2 Using the Sitecore Update Center

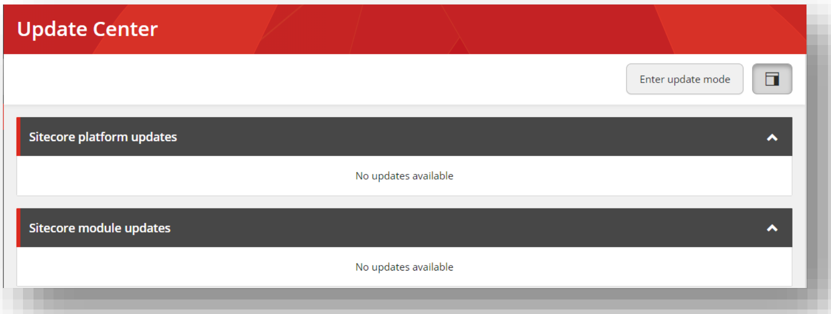
Before you can use the Update Center to install the Sitecore XP upgrade package, you must update the Update Installation Wizard. The Update Center uses the wizard to install upgrade packages.

To update the Update Installation Wizard:

1. On the **Sitecore Launchpad**, click **Control Panel**, and in the **Administration** section, click **Install a package**.
2. Do one of the following:
   - If you are upgrading from Sitecore XP 9.1.0 or later, install the `Sitecore Update Installation Wizard 4.0.1 rev.00153 for 9.1.zip` package.
   - If you are upgrading from Sitecore XP 9.0.2, install the `Sitecore Update Installation Wizard 4.0.1 rev.00153 for 8.x-9.0.zip` package.

To enter update mode:

1. On the **Sitecore Launchpad**, open the **Update Center**.
2. In **Update Center**, click **Enter update mode**. For more information, see Update mode.



Sitecore platform updates

No updates available

Sitecore module updates

No updates available

Note
Ensure that Internet Information Services (IIS) is configured to allow access to the `\sitecore\admin` folder.

Chapter 4

Upgrade Sitecore xConnect Server

If you are using the Sitecore xConnect Server and Experience Database (xDB), you must perform all the tasks described in this chapter to upgrade Sitecore xConnect Server.

This chapter describes how to upgrade xConnect from a previous version of xConnect including:

- Sitecore XP 9.0.2
- Sitecore XP 9.1 Initial Release (introduced the Sitecore Cortex™ Processing Engine role)
- Sitecore XP 9.1.1

Important
If you are upgrading from Sitecore XP 9.0.2 and want to use the new Sitecore Cortex™ Processing Engine, you must install it.

For more information about how to install and configure the Sitecore Cortex Processing Engine, see the Install the Sitecore Cortex Processing Engine chapter.

This chapter contains the following sections:

4.1 Prepare to upgrade xConnect Server

Before you upgrade xConnect, you must perform all the tasks described in this chapter.

Note
If you are going to switch from SQL Server to MongoDB as part of the upgrade, you must configure the MongoDB provider for xConnect. For information about how to configure the MongoDB provider for xConnect, see the installation guide for Sitecore XP 9.2.0.

4.1.1 Prerequisites for Upgrading xConnect

Before you upgrade xConnect:

1. Download the following files from the Sitecore Experience Platform download site:
   - `Sitecore 9.2.0 rev. 002893 (WDP XP0 packages).zip.` This file is noted as **Packages for XP Single** in the **Download options for On Premises deployment** section of the download site.
   - `Sitecore 9.2.0 rev. 002893 (upgrade files).zip`. This file is noted as **Configuration files for upgrade** in the **Upgrade options** section of the download site. You might have already been downloaded this file if you followed the procedures in the Files and scripts for upgrading section.
2. Unpack the `Sitecore 9.2.0 rev. 002893 (WDP XP0 packages).zip` file into a folder. Several files ZIP files are available in the folder.

   Unpack the `Sitecore 9.2.0 rev. 002893 (OnPrem)_xp0xconnect.scwdp.zip` file.
3. Unpack the `Sitecore 9.2.0 rev. 002893 (upgrade files).zip` file into a folder. Several files ZIP files are available in the folder.

   Unpack the `Database Upgrade Script.zip` file. This file might already be unpacked if you followed the steps in section Files and scripts for upgrading.

   It contains several xConnect database upgrade scripts including:
   - `SXP_marketingautomation.sql`
   - `SXP_referencedata.sql`
   - `SXP_collection.sql`
   - `SXP_collection_AE_Part1.sql`
   - `SXP_collection_AE_Part2.sql`
   - `SXP_collection_GrantPermissions.sql`
   - `SXP_Messaging.sql`
   - `SXP_processing_engine_tasks.sql`
   - `SXP_processing_pools.sql`
   - `SXP_mongodb_collection.js`
4. If you are using Solr provider, ensure that you have updated Solr to version 7.5.
5. If you have any functionality that accesses the contact/interaction data extraction process from outside of xConnect, you must stop it. You can restart it when the upgrade

is completed. An example of this kind of functionality is rebuilding the *Reporting* database.

Important
If you are upgrading from Sitecore XP 9.0.2, you cannot pause these tasks before the upgrade and resume them afterwards because the extraction cursors were changed in Sitecore XP 9.1.0.

If you are upgrading from Sitecore XP 9.1.0, all the cursors will still work correctly.

4.1.2 Backup your Customizations

Before you start the upgrade, backup all the customizations that you have made to the xConnect instance to ensure that they are not lost. Customizations could include:

- Configuration files
- Custom xConnect model JSON files
- Custom DLL files

You must also ensure that the default Sitecore configuration files do not contain any edits as these are overwritten during the upgrade process. If you have edited a configuration file, extract the changes to a custom configuration file which can be backed up.

Copy your customisations to a safe location to preserve the files.

4.1.3 Stop Windows services

Before you start the upgrade, stop the xConnect Windows services.

Ensure that the following Windows services are stopped:

- Sitecore XConnect Search Indexer
- Sitecore Marketing Automation Engine
- Sitecore Cortex Processing Engine

  If you are upgrading from Sitecore XP 9.1.0 or later, the *Sitecore Cortex Processing Engine* Windows service is already installed and you must stop it before the upgrade.

  Before you start the upgrade, we recommend that you:

  - Stop scheduling new processing tasks.
  - Wait for all the ongoing and pending tasks to complete.
  - Ensure that there are no more tasks in the queue.

  You can see the list of tasks in the *ProcessingEngineTasks* database, in the *Tasks* table.

  If you do not wait for the ongoing tasks to complete, when you restart the *Sitecore Cortex Processing Engine* Windows service after the upgrade:

  - Any deferred tasks that are in the *InProgress* state are re-run.
  - Any distributed tasks that are in the *InProgress* state continue from the point they were at when you stopped the Windows service.

4.2 Upgrade the xDB databases

To upgrade the databases used by xConnect, you must follow all the steps described in this section.

4.2.1 Upgrade existing 9.0.2 databases

Important
If you are upgrading from Sitecore XP 9.0.2, you must upgrade the databases as described in this section. If you are upgrading from Sitecore XP 9.1.0 or later, skip this section.

To upgrade the databases:

1. Execute the `SXP_processing_pools.sql` script for the *Xdb Processing Pools* database.
2. Execute the `SXP_marketingautomation.sql` script for the *Xdb Marketing Automation* database.
3. Execute the `SXP_referencedata.sql` script for the *Xdb Reference Data* database.

4.2.2 Upgrade the Collection database on SQL Server

Important
This section applies if the *Collection* database is on SQL Server.

If the *AlwaysEncrypted* feature *is not* configured for the *Collection* database:

1. Execute the `SXP_collection.sql` script for every shard.
2. Execute the `SXP_collection_GrantPermissions.sql` script or execute the following SQL script for every least privileged user in every shard. You must specify the user name in the script:

```
GRANT SELECT ON [xdb_collection].[UnlockContactIdentifiersIndex_Staging] TO
[collection_user]
GRANT INSERT ON [xdb_collection].[UnlockContactIdentifiersIndex_Staging] TO
[collection_user]
```

If the *Always Encrypted* feature *is* configured for the *Collection* database:

1. Execute the `SXP_collection_AE_Part1.sql` script for every shard.
2. In the new *[UnlockContactIdentifiersIndex_Staging]* table, configure deterministic encryption for the *[Identifier]* and *[Source]* columns, for every shard.

   To simplify the configuration process, SQL Server Management Studio provides an Encryption Wizard.
3. Execute the `SXP_collection_AE_Part2.sql` script for every shard.
4. Execute the `SXP_collection_GrantPermissions.sql` script or execute the following SQL script for every least privileged user in every shard. You must specify the user name in the script:

```
GRANT SELECT ON [xdb_collection].[UnlockContactIdentifiersIndex_Staging] TO
[collection_user]
GRANT INSERT ON [xdb_collection].[UnlockContactIdentifiersIndex_Staging] TO
[collection_user]
```

4.2.3 Upgrade the Collection database on MongoDB

Important
This section applies if the *Collection* database is on MongoDB.

This section only applies if you are upgrading from Sitecore XP 9.1.0. Otherwise skip this section.

If you are going to move from SQL Server to MongoDB as part of the upgrade, skip this section.

The installation guide for Sitecore XP 9.2.0 describes how to configure the MongoDB provider for xConnect.

To upgrade the *Collection* database, execute the `SXP_mongodb_collection.js` script:

```
mongo 127.0.0.1:27017/collection SXP_mongodb_collection.js
```

To build indexes efficiently, follow the suggestions in the article *Index Build Operations on a Populated Collection*.

4.2.4 Upgrade the Processing Engine Tasks database

Important
If you are upgrading from Sitecore XP 9.1.0, you must upgrade the *Processing Engine Tasks* database.

To upgrade the the *Processing Engine Tasks* database:

1. Execute the `SXP_processing_engine_tasks.sql` script for the *Processing Engine Tasks* database.
2. Run the `EXECUTE [sitecore_processing_tasks].[GrantLeastPrivilege] '<user_name>'` query - where `<user_name>` is the name of the least privilege user that exists in the *Processing Engine Tasks* database.

   This updates the permissions for the least privilege user in the *Processing Engine Tasks* database.

4.2.5 Upgrade the Messaging database

To upgrade the Messaging database, execute the `SXP_Messaging.sql` script for the *Messaging* database.

4.3 Update the Files

Update the files on the xConnect instance.

Note
You can use a Windows batch script to automate steps 1-2 in the following procedure. For more information about the batch script, see Appendix A: The xConnect Application Preparation Script.

To update the files:

1. Delete all the files under the following folders, but leave the folders themselves to ensure that the file permissions are preserved:
   - `\Bin`
   - `\App_Data\collectiondeployment`
   - `\App_Data\Config`
   - `\App_Data\Copyrights`
   - `\App_Data\Diagnostics`
   - `\App_Data\Models`
   - `\App_Data\solrcommands`
   - `\App_Data\jobs\continuous\AutomationEngine`
   - `\App_Data\jobs\continuous\AutomationEngine\App_Data\Config`
   - `\App_Data\jobs\continuous\AutomationEngine\App_Data\Diagnostics`
   - `\App_Data\jobs\continuous\IndexWorker`
   - `\App_Data\jobs\continuous\IndexWorker\Copyrights`
   - `\App_Data\jobs\continuous\IndexWorker\App_Data\Config`
   - `\App_Data\jobs\continuous\IndexWorker\App_Data\Diagnostics`
   - `\App_Data\jobs\continuous\IndexWorker\App_Data\Models`
   - `App_Data\jobs\continuous\ProcessingEngine`
   - `App_Data\jobs\continuous\ProcessingEngine\App_Data\Config`
   - `App_Data\jobs\continuous\ProcessingEngine\App_Data\Diagnostics`
   - `App_Data\jobs\continuous\ProcessingEngine\App_Data\Models`
2. In the `\App_Data\jobs\continuous\IndexWorker` folder, delete the following language folders if they exist:
   - `\de-DE`
   - `\es-ES`
   - `\fr-FR`
   - `\it-IT`
   - `\ja-JP`
   - `\ko-KR`
   - `\pt-BR`
   - `\ru-RU`
   - `\zh-CN`
   - `\zh-TW`

3. Copy all the files and folders from the `Sitecore 9.2.0 rev. 002893 (OnPrem)_xp0xconnect.scwdp\Content\Website` folder that you unpacked earlier, with the exception of the following folders:
   - `\App_Config`
   - `\App_Data\jobs\continuous\AutomationEngine\App_Config`
   - `\App_Data\jobs\continuous\IndexWorker\App_Config`
   - `\App_Data\jobs\continuous\ProcessingEngine\App_Config`
   - `\App_Data\logs`
4. When prompted, replace any files that have the same name.

4.3.1 Add New Connection Strings

Important
If you are upgrading from Sitecore XP 9.1.0 or later, these connection strings are already added and you can skip this section.

1. Add the reporting connection string to the xConnect instance.

   In the `xconnect\App_Config\ConnectionStrings.config` file, add a new connection string for the *Reporting* database:

   ```
   <add name="reporting" connectionString="user id=user;password=password;Data
   Source=(server);Initial Catalog=Sitecore.Reporting" />
   ```

2. Add the reporting.client connection strings to the Sitecore instance.

   In the `sitecore\App_Config\ConnectionStrings.config` file, add a new connection string for the *Sitecore.Reporting.Client*:

   ```
   <add name="sitecore.reporting.client" connectionString="https://Sitecore_xconnect"/>
   <add name="sitecore.reporting.client.certificate"
   connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;Fi
   ndValue="/>
   ```

4.3.2 Restore Customizations

You can now reapply the customizations that you backed up earlier.

To reapply your customizations, copy the backup files back to their original locations.

4.4 Configure the Search Provider

You must now configure the search provider that you have chosen to use.

4.4.1 Configure Solr

Solr is the default search engine for content search and xConnect search for on-premise installations.

If you installed a new instance of Solr instead of updating the existing installation to Solr 7.5, you must configure the new Solr cores. You must also apply schema changes to the existing xConnect Solr cores.

If you are upgrading to Solr 7.5, skip this section. You will need to review the Upgrading Solr section in the Configure Sitecore chapter.

To create new Solr cores, in the Solr admin console, add two new Solr cores that are based on the default configuration and use the same base name, for example:

- *xdbcore*
- *xdbcore_rebuild*

Important
One of the cores *must* be called *<custom_name>_rebuild.*

1. Use the Solr Schema API to apply the schema changes from the `xconnect\App_data\solrcommands\schema.json` file for both cores.

Here is an example of a PowerShell script that uses the Solr Schema API to apply these changes:

```
$pathToXdbSchemaFile="<path_to_instance>\xconnect\App_data\solrcommands\schema.json"
$apiUrl="https://<host>:<port>/solr/<custom_name>/schema"

$requestBody = Get-Content $pathToXdbSchemaFile -Raw | ConvertFrom-Json | ConvertTo-Json
Invoke-WebRequest -Uri $apiUrl -ContentType "application/json" -Method Post -
UseBasicParsing -Body $requestBody | ConvertFrom-Json | ConvertTo-Json
```

2. In the following connection string files:

- `xconnect\App_Config\ConnectionStrings.config`
- `xconnect\App_data\jobs\continuous\IndexWorker\App_config\ConnectionStrings.config`

Update the `solrCore` connection string with the name of the core that is attached to Solr 7.5:

```
<add name="solrCore" connectionString="https://<host>:<port>/solr/<custom_name>"/>
```

4.4.2 Configure Azure Search

To configure Azure Search:

1. In the `xconnect\App_data\config\sitecore\CollectionSearch` folder:

   Enable the following configuration file by removing the `.disabled` extension:

   - `sc.Xdb.Collection.IndexWriter.AzureSearch.xml.disabled`

   Disable the following configuration files by adding the `.disabled` extension:

- `sc.Xdb.Collection.IndexReader.SOLR.xml`
- `sc.Xdb.Collection.WebClient.SOLR.xml`

2. In the `xconnect\App_data\jobs\continous\IndexWorker\App_data\config\sitecore` folder:

   Enable the following configuration files by removing the `.disabled` extension:

   - `\SearchIndexer\sc.Xdb.Collection.IndexWriter.AzureSearch.xml.disabled`
   - \`CollectionSearch\sc.Xdb.Collection.IndexWriter.AzureSearch.xml.disabled`

   In Sitecore XP 9.0.2 and later versions, in the `\CollectionSearch\sc.Xdb.Collection.IndexWriter.AzureSearch.xml.disabled` file, the `DataReplicationTimeoutMilliseconds` setting is disabled by default. If you use more than one replica, you must enable this setting.

3. In the `xconnect\App_data\jobs\continous\IndexWorker\App_data\config\sitecore` folder, disable the following configuration files by adding the `.disabled` extension:

   - `\CollectionSearch\sc.Xdb.Collection.IndexReader.SOLR.xml`
   - `\CollectionSearch\sc.Xdb.Collection.WebClient.SOLR.xml`
   - `\SearchIndexer\sc.Xdb.Collection.IndexWriter.SOLR.xml`

4. In the following connection string files:

   - `xconnect\App_Config\ConnectionStrings.config`
   - `xconnect\App_data\jobs\continuous\IndexWorker\App_config\ConnectionStrings.config`

   Add (uncomment) and update the Azure Search connection string, the default name is collection.search.

```
<add name="collection.search"
connectionString="serviceUrl=https://[service].search.windows.net/;indexName=[index
name];apiKey=[API Key]"/>
```

4.5 Configure Azure Key Vault

In Sitecore XP, you can protect sensitive data with *Always Encrypted* and store encryption keys in Azure Key Vault.

To configure Azure Key Vault:

1. Enable the `xconnect\App_Data\Config\Sitecore\Collection\sc.Xdb.Collection.Data.Sql.KeyStoreProviders.xml` file by renaming the `sc.Xdb.Collection.Data.Sql.KeyStoreProviders.xml.disabled` file to `sc.Xdb.Collection.Data.Sql.KeyStoreProviders.xml`.

2. In the `xconnect\App_Data\Config\Sitecore\Collection\sc.Xdb.Collection.Data.Sql.KeyStoreProviders.xml` file, replace `<ClientIdAppSettingsKey>` and `<ClientSecretAppSettingsKey>` values with the appropriate values from the `xconnect\App_Data\Config\Sitecore\Collection\sc.Xdb.Collection.Data.Sql.KeyStoreProviders.xml` file.

   Note
   If you are using the default names for the applications settings and you have no custom code that requires these settings, you can use the new default names of the application settings `AzureKeyVaultClientId` and `AzureKeyVaultClientSecret` in the `xconnect\ App_Config\AppSettings.config` file.

3. Disable the `xconnect\App_Data\Config\Sitecore\Collection\sc.Xdb.Collection.Data.Sql.KeyStoreProviders.xml` file by adding `.disabled` to the file extension.

For more information about Azure Key Vault, see Microsoft's documentation.

Chapter 5

Install the Sitecore Cortex Processing Engine

If you want to use the Sitecore Cortex™ Processing Engine service in Sitecore XP 9.2.0, you must install and configure it.

The Sitecore Cortex Processing Engine is a new role that was introduced in Sitecore XP 9.1.0 that you can use with or without Microsoft Machine Learning Server (MLS).

Important
If you are upgrading from Sitecore XP 9.1.0 or later with xConnect installed, you can skip this chapter as the Sitecore Cortex Processing Engine is already installed.

If you are upgrading from Sitecore XP 9.0.2, you must install and configure the Sitecore Cortex Processing Engine. This chapter explains how to install and configure the Sitecore Cortex Processing Engine.

Alternatively Sitecore Cortex Processing Engine can be installed with Sitecore xConnect if not already installed. For information about installing Sitecore xConnect, see the Sitecore Experience Platform 9.2.0 Installation Guide.

For more detailed information about installing the Sitecore Cortex Processing Engine — including the deployment topologies — see the Sitecore Experience Platform 9.2.0 Installation Guide.

This chapter contains the following sections:

5.1 Install and Configure Sitecore Cortex Processing Engine

This section describes how to install and configure Sitecore Cortex Processing Engine for a single instance Sitecore solution.

Before you install Sitecore Cortex Processing Engine service:

Acquire resources

1. Download the following file from the Sitecore Experience Platform download site:
   - `Sitecore 9.2.0 rev. 002893 (WDP XP0 packages).zip`. This file is noted as **Packages for XP Single** in the **Download options for On Premises deployment** section of the download site.
   - You might have already been downloaded this file if you followed the procedures in the Files and scripts for upgrading section.
2. Unpack the `Sitecore 9.2.0 rev. 002893 (WDP XP0 packages).zip` file into a folder. Several files ZIP files are available in the folder.
3. Unpack the `Sitecore 9.2.0 rev. 002893 (OnPrem)_xp0xconnect.scwdp.zip` file.

Deploy databases

Deploy the following files to SQL server:

- `Sitecore.Processing.engine.storage.dacpac`
- `Sitecore.Processing.engine.tasks.dacpac`

Copy files and update Configuration

1. Copy the following folder:

   `\Sitecore 9.2.0 rev. 002893 (OnPrem)_xp0xconnect.scwdp\Content\Website\App_Data\jobs\continuous\ProcessingEngine`
2. Paste into the `C:\inetpub\wwwroot\instanceName.xconnect\App_Data\jobs\continuous` folder.
3. In the `instanceName.xconnect\App_data\jobs\continuous\ProcessingEngine\App_Config\ConnectionStrings.config` file, update the following connection strings:
   - `processing.engine.storage`
   - `processing.engine.tasks`
   - `processing.webapi.blob`
   - `processing.webapi.table`
   - `xconnect.collection`
   - `xconnect.collection.certificate`
   - `xconnect.configuration`
   - `xconnect.configuration.certificate`
   - `xconnect.search`
   - `xconnect.search.certificate`

- `messaging`
- `reporting`

The remaining connection strings are related to the optional integration with MLS server and are described later in this document.

> **Note**
> The `processing.webapi.blob` and `processing.webapi.table` APIs are deployed with your xConnect instance and should point to your xConnect endpoint for example https://sc920.xconnect
>
> In a scaled environment the `processing.webapi.blob` and the `processing.webapi.table` connection strings should point to the new *Processing Engine* role.

4. In the `xconnect\App_data\jobs\continuous\ProcessingEngine\App_Data\` folder, add a valid `license.xml` file
5. In the `xconnect\App_data\jobs\continuous\ProcessingEngine\App_Data\` folder, create a `Logs` folder and grant the *NT AUTHORITY\LOCAL SERVICE* security group *Full control* permission to it.
6. To install the Sitecore Cortex Processing Engine to run as a Windows Service, run the Command Prompt with *Administrator* permissions, execute the `sc create` command, and pass the appropriate parameters:

```
sc create instanceName.xconnect-ProcessingEngineService binPath=
"C:\inetpub\wwwroot\instanceName.xconnect\App_Data\jobs\continuous\ProcessingEngine\Si
tecore.ProcessingEngine.exe -service" start= auto obj= "NT AUTHORITY\LOCAL SERVICE"
DisplayName= "Sitecore Cortex Processing Engine - instanceName.xconnect-
ProcessingEngineService"
```

Add the *Processing Engine Storage* database connection string to the xConnect instance

In the `xconnect\App_Config\ConnectionStrings.config` file, add a new connection string for the *Processing Engine Storage* database:

```
<add name="processing.engine.storage" connectionString="user id=user;password=password;Data
source=(server);Initial Catalog=ProcessingEngineStorage" />
```

5.2 Install and Configure Microsoft Machine Learning Server

After you have configured the Sitecore Cortex Processing Engine, you can install and configure the optional Microsoft Machine Learning Server (MLS) integration.

Note
There are no worker processes in the Sitecore Cortex Processing Engine that use the MLS integration by default.

For more information about how to install Microsoft Machine Learning Server see the Sitecore Experience Platform 9.2.0 Installation Guide.

Chapter 6

Upgrading Sitecore XP

You use the Sitecore Update Center to upgrade the Sitecore platform and modules. Before you use the Update Center to upgrade to Sitecore 9.2.0, you must have all the prerequisite that are described in this guide in place.

Note
Before you install the upgrade package, you must make a backup of your website.

This chapter contains the following sections:

- Using the Update Center to Upgrade to Sitecore 9.2.0
- Troubleshooting

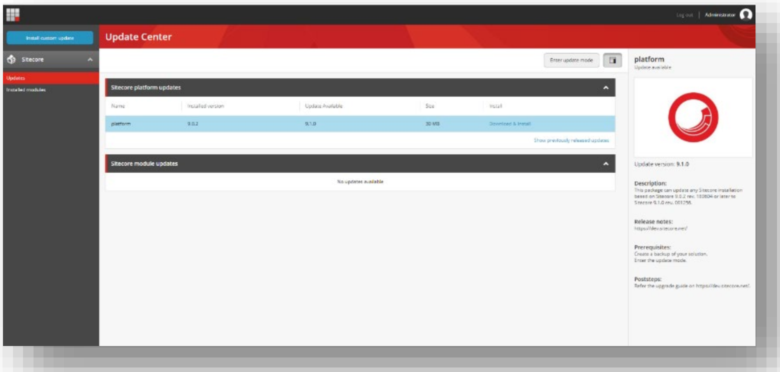
6.1 Using the Update Center to Upgrade to Sitecore 9.2.0

6.1.1 Install the Upgrade Package

You can now use the Update Center to install the upgrade package. Before doing so, ensure you have followed the preparation steps described in chapter *Preparing to Upgrade*.

To install the upgrade package:

1. Log in to the **Launch Pad** and open the **Update Center**.



Name	Installed version	Update Available	Size	Install
platform	9.0.2	9.1.0	30 MB	Download & Install

No updates available

platform
Update available

Update version: 9.1.0

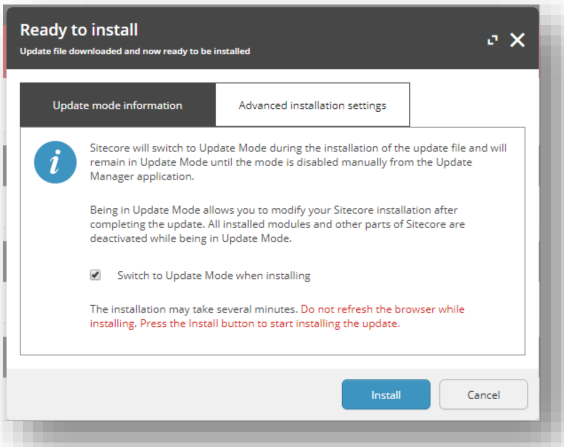
Description:
This package can update any Sitecore installation based on Sitecore 9.0.2 rev. 180604 or later to Sitecore 9.1.0 rev. 001256.

Release notes:
https://dev.sitecore.net/

Prerequisites:
Create a backup of your solution.
Enter the update mode.

Poststeps:
Refer the upgrade guide on https://dev.sitecore.net/.

2. Select the Sitecore XP 9.2.0 upgrade package and then click **Download and Install**.
3. When the package is downloaded, in the **Ready to install** dialog box, click **Install**:


Ready to install
Update file downloaded and now ready to be installed
Update mode information
Advanced installation settings
Install
Cancel

Sitecore will switch to Update Mode during the installation of the update file and will remain in Update Mode until the mode is disabled manually from the Update Manager application.

Being in Update Mode allows you to modify your Sitecore installation after completing the update. All installed modules and other parts of Sitecore are deactivated while being in Update Mode.

☑ Switch to Update Mode when installing

The installation may take several minutes. Do not refresh the browser while installing. Press the Install button to start installing the update.

4. Wait until the package is installed.

After the upgrade package has been installed, you must configure Sitecore.

6.2 Troubleshooting

This section describes how to deal with some of the issues that can occur during the upgrade process.

I can not see an update to Sitecore XP 9.2.0 in the Update Center

Check the error message that appears in the Update Center.

When I click Install, the installation fails

Before installing the items and files, the Sitecore instance goes into update mode and disables the majority of the configuration files and leaves only those files that are needed to install the Update. If the Sitecore instance failed to enter update mode, the installation cannot proceed and your instance cannot start.

This also means that the default Sitecore configuration was changed directly and not by using the patching functionality. Check the log files and revert any direct changes that were made to the Sitecore configuration files. If you still experience problems, raise a support ticket.

The installation starts successfully but fails when I trying to install the package

The update process is divided into three steps:

- Installing the items.
- Installing the files.
- Performing the post upgrade steps.

In the **Ready to install** dialog, in the **Advanced installation settings** tab, try to install the items first and verify that they were installed without any errors. Then try to install the files and finally perform the post upgrade steps. If you still experience problems, raise a support ticket.

When I click Exit the update mode, my Sitecore instance fails to start

When you exit update mode, the entire configuration is applied to your Sitecore instance.

Read the error message that you see and check that all the configuration nodes are present and enabled.

Ensure you have recompiled your solution with the updated Sitecore dll's.

My Sitecore instance is unstable after being updated

Check the log files to see if there are any exceptions.

Check that you have performed all the post upgrade steps without receiving any exceptions.

Ensure that you have setup any additional services that are required for your Sitecore XP 9.2.0 instance to run.

Chapter 7

Configure Sitecore

After you have installed the Sitecore upgrade package, you must review the custom changes in the configuration files and perform the procedures described in this chapter.

This chapter contains the following sections:

7.1 Review Changes in Configuration Files

7.1.1 The Helix Folder Structure

Sitecore XP 9.1.0 or later creates the default Helix folder structure. The `Website\App_Config\Include` folder contains `Foundation`, `Feature`, and `Project` folders, which are included in the `Layers.config` file with the appropriate load order.

This section applies if you are upgrading from Sitecore 9.0.2. Skip this section if you are upgrading from Sitecore 9.1.0 or later.

Note
The load order of the Helix folder takes precedence over the file load order because it is included in the `Layers.config` file.

The folders and files are loaded in the following order:

- Foundation
- Feature
- Project
- The files in the `Website\App_Config\Include` folder.

If you need certain files to load first, you must place them in the `Website\App_Config\Include\Foundation` folder.

7.1.2 Review patch files with customization

After installing the upgrade package, you must enable the patch files with the customizations that were generated by the **Update Installation Wizard**.

The patch files are created and stored in the same folder as the original configuration files. They are saved in the following file format: `<filename>.patch.config.disabled`.

To facilitate traceability and debugging, the wizard also saves the original file with your customizations in the following file format: `<filename>.custom.config.disabled`.

For example, if you customized the `Sitecore.Mvc.config` file, during the analysis, the **Update Installation Wizard** creates:

- `Sitecore.Mvc.patch.config.disabled`

  A patch file with your customizations.

- `Sitecore.Mvc.custom.config.disabled`.

  A backup of the customized configuration file.

  Both of these files are disabled.

To enable the patch files with the customizations that were generated by the **Update Installation Wizard**, remove the `.disabled` extension from the file name.

Note
The best practice is to store all the generated patch files in the `\Website\App_Config\Include\Z.Custom` folder.

7.1.3 Review the Web.config Customizations

Sometimes the **Update Installation Wizard** cannot generate the configuration file patches, for example, when there are non-Sitecore sections in the `web.config` file. When this happens, the **Update Installation Wizard** extracts the customizations from the configuration file, and merges the customizations with the corresponding file from the upgrade package.

The merged configuration file is then saved as `<filename>.merged.config.disabled` and the customized configuration file is saved as `<filename>.custom.config.disabled`.

To enable your customizations:

1. Rename the `<filename>.config` file, to `<filename>.config.old`.
2. Review and rename the merged file `<filename>.merged.config.disabled` to `<filename>.config`.

Note
Any customizations in the `web.config` file are stored in the `Web.merged.config.disabled` file after upgrading Sitecore.

7.1.4 Reviewing and specifying the search provider

Since Sitecore 9.0, the search provider is specified in the `Web.config` file. After upgrading to Sitecore 9.2.0 the search provider is automatically set to `Solr` in the updated `Web.config` file.

The original version of the `Web.config` file that was used before the Sitecore upgrade was started is saved as `Web.merged.config.disabled`. All prior customizations are stored in this file.

If your search provider before upgrading was:

- **Solr** – skip the rest of this section.
- **Azure** – modify the `Web.config` file and specify Azure as the search provider.
- **Lucene** – modify the `Web.config` file and specify Lucene as the search provider.

  Consider implementing Solr or Azure as your search provider as support for Lucene is deprecated in Sitecore Experience Platform 9.2.0 and will be completely removed in a later release.

To specify the search provider:

- In the `\Website\Web.config` file, in the `<AppSettings>` section, specify the search provider:

  ```
  <add key="search:define" value="[search provider]" />
  ```

  The following values are supported:

  - Lucene - support for Lucene is deprecated in Sitecore Experience Platform 9.2.0 and will be completely removed in a later release.
  - Solr - default value is *Solr* as of Sitecore 9.2.
  - Azure - the Azure Search provider is only supported for Azure Cloud PaaS deployments.

7.1.5 Reviewing and specifying the server role

Sitecore 9.0 introduced the new rule-based configuration functionality that you use to specify the server role performed by each Sitecore instance in your solution. With this functionality, you no longer have to disable and enable configuration files to configure a Sitecore role.

Starting with Sitecore XP 9.0, the `Web.config` file contains the `role:define` setting that defines the Sitecore server role.

After upgrading to Sitecore 9.2 the server role is automatically set to `Standalone` in the updated `Web.config` file. The original version of the `Web.config` file before the Sitecore upgrade was started is saved as `Web.merged.config.disabled`. All prior customization are stored in this file.

If the server role before upgrading was:

- **Standalone** – skip the rest of this section.
- **Any other setting** – modify the `Web.config` file and re-specify the server role to the original value before the upgrade.

Note
You must specify the server role after you have upgraded each Sitecore instance.

To specify the server role:

- In the `\Website\web.config` file, in the `<AppSettings>` section, specify the relevant server role:

  ```
  <AppSettings>
   <add key="role:define" value="[server role]"/>
  </AppSettings>
  ```

  The following server roles are supported:

  - ContentDelivery
  - ContentManagement
  - ContentDelivery, Indexing (introduced in Sitecore XP 9.2.0)
  - ContentManagement, Indexing (introduced in Sitecore XP 9.2.0)
  - Processing
  - Reporting
  - Standalone (default)

  The *Standalone* default value means that one Sitecore instance performs all the Sitecore server roles.

New Indexing sub-role

Sitecore XP 9.2.0 introduces a new Indexing sub-role to the Sitecore server roles to simplify the configuration of content indexing.

When you assign the Indexing sub-role to a server role, that role manages content indexing automatically. The other server roles that have not been assigned this sub-role do not index content.

You configure the Indexing sub-role in the web.config file and can combine it with the CM and CD roles, for example, "ContentManagement, Indexing".

If you have multiple CM or CD servers, you should only assign this sub-role to one of the servers.

7.1.6 Update Sitecore.Analytics.ExcludeRobots.config file

In website\App_Config\Sitecore\Marketing.Tracking\Sitecore.Analytics.ExcludeRobots.config remove the user agents from <excludedUserAgents>.

Sitecore uses the device detection component to identify the robots and the user agent previously in this list are detected as robots.

If you have additional user agents added to the list you can you can check whether user agent is known as bot at https://51degrees.com/resources/user-agent-tester. If the user agent is known as a robot you can safely remove this from the list.

7.2 Upgrading Solr

To continue using the Solr search provider in Sitecore XP 9.2.0, you must upgrade your existing Solr instance to version 7.5. After upgrading to Solr 7.5, you can create new cores with names that correspond to the Solr version.

For more information on how to set up Solr for use with Sitecore, see https://doc.sitecore.com/developers/92/platform-administration-and-architecture/en/walkthrough--setting-up-solr.html.

For more information on how to set up SolrCloud for use with Sitecore, see https://doc.sitecore.com/developers/92/platform-administration-and-architecture/en/walkthrough--setting-up-solrcloud.html.

7.2.1 Create new Solr indexes

To create new indexes:

1. Open the Solr folder. Depending on your Solr installation, the location of this folder might be different. The default location is: `C:\solr-7.5.0\server\solr\`
2. Copy the Solr provided configuration example from the `C:\solr-7.5.0\server\solr\configset\_default` folder.
3. Paste the example into `Solr/server/solr` folder and rename it with the appropriate index name - `C:\solr-7.5.0\server\solr\sitecore_core_index`
4. In the folder, create an empty file and name it `core.properties`.
5. Edit the `core.properties` file and add:

```
name=[CORE_NAME]
config=solrconfig.xml
update.autoCreateFields=false
dataDir=data
```

6. Open the `C:\solr-7.5.0\server\solr\sitecore_core_index\conf\managed-schema` file.
7. In the `<uniqueKey>id</uniqueKey>` node, set the `uniqueKey` value to `_uniqueid`:

```
<uniqueKey>_uniqueid</uniqueKey>
```

8. In the `<field>` section, add the field configuration for the `_uniqueid` value:

```
<field name="_uniqueid" type="string" indexed="true" required="true" stored="true"/>
```

9. Save your changes.
10. If you use a new name for the index, you must update the corresponding setting in the config file.
11. Repeat all these step for each index.

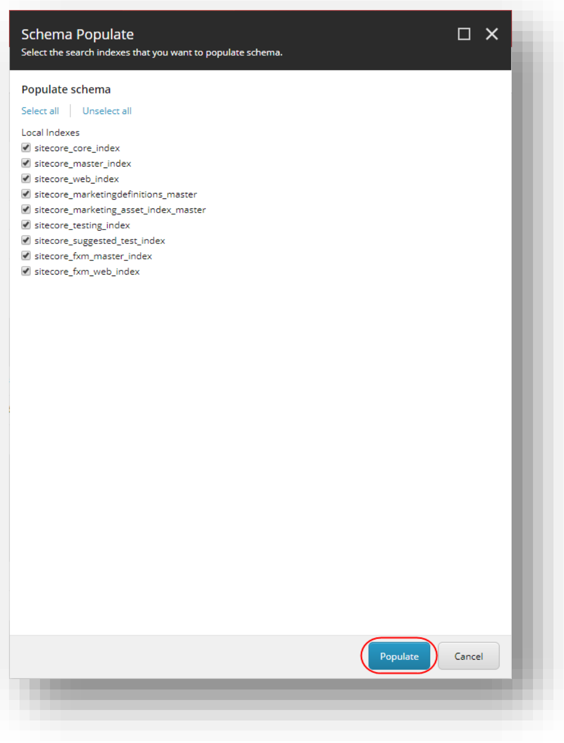
7.2.2 Populate the Solr Managed Schema

Important
During the installation of the update package, the previous configuration file is disabled. If the file has any cuztomizations, a new merged file is created and disabled. If you have any cuztomizations, you should disable the new configuration file, review and enable merged file. For more information, see the section Review Changes in Configuration Files.

You must populate the managed schema before Solr can use it.

To populate the Solr managed schema:

1. On the **Sitecore Launchpad**, click **Control Panel**, and in the **Indexing** section, click **Populate Solr Managed Schema**.
2. In the **Schema Populate** dialog box, select all indexes, and then click **Populate**.



Schema Populate

Select the search indexes that you want to populate schema.

Populate schema

Select all | Unselect all

Local Indexes

☑ sitecore_core_index
☑ sitecore_master_index
☑ sitecore_web_index
☑ sitecore_marketingdefinitions_master
☑ sitecore_marketing_asset_index_master
☑ sitecore_testing_index
☑ sitecore_suggested_test_index
☑ sitecore_fxm_master_index
☑ sitecore_fxm_web_index

Chapter 8

Post-upgrade Steps

After installing the upgrade package and upgrading the modules, you must complete some post-upgrade steps.

This chapter contains the following sections:

8.1 General Maintenance

After you finalize the upgrade and upgrade your modules, and before you start using Sitecore, you must:

- Clear the browser cache.
- Republish your website.

  You must publish the entire site, including system items, templates, and so on, to every publishing target.
- Rebuild the search indexes.
  1) On the **Sitecore Launchpad**, click **Control Panel**, and in the **Indexing** section, click **Indexing manager**.
  2) In the **Indexing Manager** dialog box, click **Select all**, and then click **Rebuild**.
- Rebuild the *Link* database in the *Master* and *Core* databases.
  1) On the **Sitecore Launchpad**, click **Control Panel**, and in the **Database** section, click **Rebuild link databases**.
  2) Select the *Master* and *Core* databases and then click **Rebuild**.
- Clean up all the databases
  1) On the **Sitecore Launchpad**, click **Control Panel**, and in the **Database** section, click **Clean up databases**.
  2) Select all the databases and then click **Clean**.
- Import client languages

  If you had been using a client language other than English before the upgrade, you must download and import the latest translations from the *Client Translations* section of the Download site. Client translations are available in:

  - Danish (da-DK)
  - German (de-DE)
  - Japanese (ja-JP)
  - Chinese (zh-CN)

  For more information about how to apply a client language, see the Sitecore Experience Platform Client Translations Manual.

8.1.1 Update personalization rules

Note
This section applies if you are upgrading from Sitecore 9.0.2. If you are upgrading from Sitecore 9.1.0 or later, skip this section.

In Sitecore XP 9.1.0, the following personalization conditions were modified:

- Campaign was Triggered
- Entry Page

- Day of Week
- Device Type
- Matches Pattern

If you have implemented any personalization rules that use these conditions, you *must* update them.

The following table lists the changes:

Sitecore XP 9.0 Update 2 or earlier	**Sitecore XP 9.1 or later**
where the *specific* campaign was triggered during the current visit	where at least one of campaigns *in list* was triggered during the current visit
where the Visit started at *the Entry Page*	where the Visit started at one of pages *in list*
when the current day is *day of the week*	when the current day is one of *days of the week*
where device type is *value*	where device type is one of *the list*
where the current contact matches the *specific* pattern card in the *specific* profile	where the current contact matches at least one of the profile pattern card *in lists*

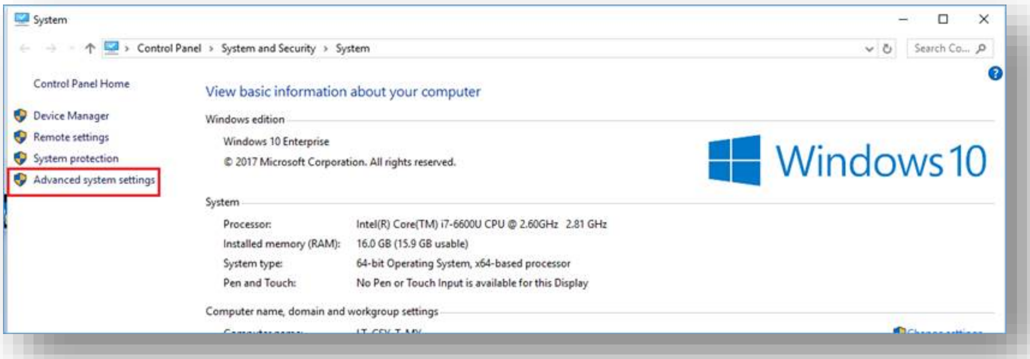
8.2 Enable Telemetry Reporting in a Production Environment

The Sitecore telemetry reporting feature gathers information that helps Sitecore understand how customers use our products. The environment type (production or non-production) helps us to associate the features used with the appropriate use-cases.

The telemetry system assumes Sitecore is running in a non-production environment by default. When you set up a production environment, you must therefore define a system variable to report the environment type.

To define a system variable:

1. In the Control Panel, click **System**, **Advanced system settings**.



View basic information about your computer

Windows edition

Windows 10 Enterprise

© 2017 Microsoft Corporation. All rights reserved.

System

Processor: Intel(R) Core(TM) i7-6600U CPU @ 2.60GHz 2.81 GHz

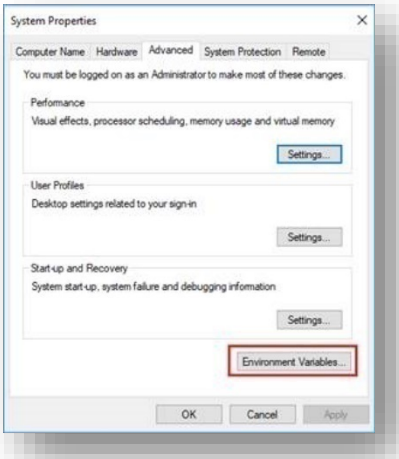
Installed memory (RAM): 16.0 GB (15.9 GB usable)

System type: 64-bit Operating System, x64-based processor

Pen and Touch: No Pen or Touch Input is available for this Display

Computer name, domain and workgroup settings

2. In the **System Properties** dialog, click **Environment Variables**.



You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage and virtual memory

User Profiles

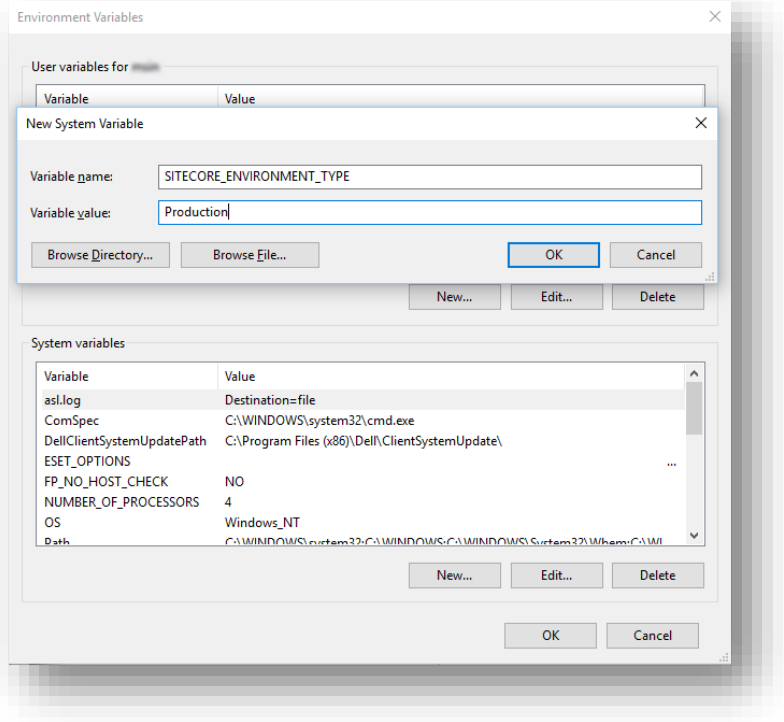
Desktop settings related to your sign-in

Start-up and Recovery

System start-up, system failure and debugging information

3. In the **Environment Variables** dialog, in the **System variables** section, click **New**.

4. In the **New System Variable** dialog, in the **Variable name** field, enter SITECORE_ENVIRONMENT_TYPE and in the **Variable value** field, enter *Production*.



Variable name: SITECORE_ENVIRONMENT_TYPE

Variable value: Production

Variable	Value
asl.log	Destination=file
ComSpec	C:\WINDOWS\system32\cmd.exe
DellClientSystemUpdatePath	C:\Program Files (x86)\Dell\ClientSystemUpdate\
ESET_OPTIONS	...
FP_NO_HOST_CHECK	NO
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WI...

5. Restart the computer.

8.3 xDB Maintenance

To continue using Sitecore xDB, perform the steps described in this section.

8.3.1 Enable xConnect services

You must start the xConnect services that you disabled in the Upgrade Sitecore xConnect chapter. These services must be restarted before exiting Update Mode in Upgrade Center.

To enable xConnect services:

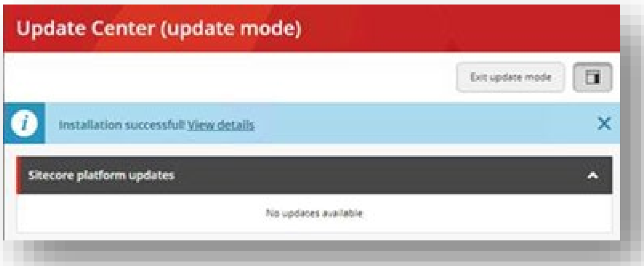
1. Recycle the IIS xConnect site Application Pool.
2. Start the *Sitecore xConnect Search Indexer* Windows service.
3. Start the *Sitecore Marketing Automation Engine* Windows service.
4. Start the *Sitecore Cortex Processing Engine* Windows service.
5. Rebuild the xConnect indexes.
6. Ensure that all xConnect services are running as expected by reviewing exceptions in the log files.

8.3.2 Exiting Update Mode

To complete the upgrade process, you must exit out of Update Mode in the Upgrade Center. Exiting from Update Mode also enables Sitecore xDB.

To exit Update Mode:

1. Log in to the **Sitecore LaunchPad** and open the **Update Center**.
2. After you have configured Sitecore, in the **Update Center**, click **Exit update mode**.



Installation successful! View details

No updates available

Your Sitecore solution is reloaded and xDB along with configuration changes are enabled.

8.3.3 Using Email Experience Manager (EXM)

You can use Sitecore EXM to create tailored email campaigns for specific segments and make them both personal and relevant to your customers.

EXM is enabled by default. To enable or disable EXM, modify the `Web.config` file in the `\Website` folder.

The `<AppSettings>` section in this file contains the following setting:

```
<add key="exmEnabled:define" value="yes" />
```

Set `value` to `no` to turn off EXM.

Important
You can only use EXM when xDB is enabled.

Define the Email Delivery Provider

To use EXM, you must define the email delivery provider used by your solution.

To define the email delivery provider:

- In the `\Website\Web.config` file, in the `<AppSettings>` section, set the email delivery provider:

  ```
  <add key="eds:define" value="CustomSmtp" />
  ```

  The following values are supported:

  - CustomSmtp
  - EmailCloud

  The default value is *CustomSmtp.*

For more information about email delivery, see Sitecore's documentation.

Enable the Transport Layer Security Protocol

You must enable the Transport Layer Security (TLS) protocol version 1.2 on all of your Sitecore XP CM and Dedicated Dispatch servers (DDS).

For more information about enabling TLS 1.2, see Microsoft's documentation.

8.3.4 Redeploy the Marketing Definitions

To redeploy the marketing definitions:

1. On the **Sitecore Launchpad**, click **Control Panel**, **Analytics**, and then click **Deploy Marketing Definitions**.
2. In the **Deploy marketing definitions** dialog box, select all the definitions and taxonomies and click **Deploy**.

Note
If you do not do this, the binary data stored in the **Data** column in the definition tables (*CampaignActivityDefinitions*, *GoalDefinitions,* and so on) might not be compatible with this version of Sitecore.

8.3.5 Upgrade the Session database

If you are using an out of process session state provider, you must:

1. Stop all the Sitecore XP instances that use the same *Session* database.
2. If you are using the MongoDB session state provider, drop the *Session* database.
3. If you are using the SQL Server session state provider, replace the *Session* database with a clean one from the Sitecore XP installation package.

4. If you are using the Redis cache, clear the cache using a Redis command such as FLUSHDB or FLUSHALL.
5. Start all the Sitecore XP instances.

8.4 Upgrading from Sitecore 9.0.2

If you are upgrading from Sitecore 9.0.2, you must perform some of the tasks described in this section.

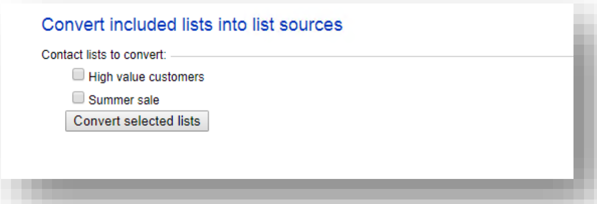
Important
If you are upgrading from Sitecore XP 9.1.0 or later, skip this section.

8.4.1 Upgrade the Included Lists in List Manager

In Sitecore XP 9.0 or earlier, you could add contact lists to other lists.

Starting from Sitecore XP 9.1.0, included lists are treated as list sources and this allows you to track contacts from different sources.If you have contact lists that contain included lists and want them to be shown as list sources, you must:

1. On the Content Management instance, open the `/sitecore/admin/ConvertContactListSources.aspx` page.



Convert included lists into list sources

Contact lists to convert:

□ High value customers

□ Summer sale

2. Select the lists that you want to convert into lists sources (only lists that contain included lists that are not treated as list sources appear on a screen) and click **Convert selected lists**

   When you open this page again, it displays the lists that are currently being processed. When all the included lists have been converted to list sources, the page is empty.

8.4.2 Upgrade the Experience Analytics Data

After you upgrade to Sitecore XP 9.2.0, you must upgrade the Experience Analytics data.

To upgrade the Experience Analytics data:

1. Locate the SXP_reporting_migrate_experience_analytics_data.sql file.

   This file is available in the `Database Upgrade Script.zip` file. This file might already be unpacked if you followed the steps in section Files and scripts for upgrading.

2. In SQL Server, select the *Reporting* database in Sitecore 9.0.x, and run the `SXP_reporting_migrate_experience_analytics_data.sql` script.

   This script deletes the following database artifacts:

   - `[dbo].Add_DeviceMetrics_Tvp`
   - `[dbo].Add_GeoMetrics_Tvp`
   - `[dbo].DeviceMetrics_Type`
   - `[dbo].GeoMetrics_Type`
   - `[dbo].Fact_DeviceMetrics`

- `[dbo].Fact_GeoMetrics`

It also migrates all of the data in the `Fact_DeviceMetrics` and `Fact_GeoMetrics` tables to other tables in the database.

8.5 Recompile your Solution

Recompile your solution with the updated Sitecore DLLs and fix any breaking changes.

Your should have tested customizations already on a test system prior to deploying your solution in production.

Chapter 9

Sitecore Identity

Sitecore Identity (SI) is a mechanism for logging in to Sitecore. It was introduced in Sitecore 9.1. It builds on the Federated Authentication functionality introduced in Sitecore 9.0 and the Sitecore Identity server, which is based on IdentityServer4. It provides a separate identity provider, and makes it possible for you to set up SSO (Single Sign-On) across Sitecore services and applications.

If you are upgrading from Sitecore 9.0.2, you must install Sitecore Identity Server and configure the Sitecore Identity Client.

If you are upgrading from Sitecore 9.1.0 or later, you must upgrade Sitecore Identity Server.

This chapter contains the following sections:

9.1 Installing Sitecore Identity Server

If you have upgraded from Sitecore 9.0.2, you must install Sitecore Identity Server and configure the Sitecore Identity Client.

You must use the latest version of Sitecore Installation Framework (SIF) to install Sitecore Identity Server. For more information about how to use SIF, see the Installation Guide.

Before you install Sitecore Identity Server, you must install .NET Core 2.1.3 Runtime or later.

To install Sitecore Identity Server:

1. Create a folder called `c:\resourcefiles`.
2. Download and save the following WDP packages and SIF resource files in this folder:
   - o `Sitecore.IdentityServer 3.0.0 rev. 00211 (OnPrem)_identityserver.scwdp.zip`

     The 2.1.1 file is available from the Sitecore Identity Download page.
   - `IdentityServer.json`
   - `createcert.json`
3. Save your Sitecore license file in the `c:\resourcefiles` folder as `license.xml`.
4. Edit the highlighted sections in the following script:

```
$prefix="sc"
$CMurl="https://$prefix"
$IdentityServer="$prefix-IdentityServer"
$ClientSecret="Random_STRING"
$SqlServer = ".\SQLEXPRESS"
$SqlAdminUser = "User"
$SqlAdminPassword="Password"
$PSScriptRoot = "c:\resourcefiles"

#install client certificate for IdentityServer
$certParamsForIdentityServer = @{
 Path = "$PSScriptRoot\createcert.json"
 CertificateName = $IdentityServer
}
Install-SitecoreConfiguration @certParamsForIdentityServer -Verbose

#install Sitecore Identity Server
    $identityParams =
    @{
        Path = "$PSScriptRoot\IdentityServer.json"
        Package = "$PSScriptRoot\Sitecore.IdentityServer 3.0.0 rev. 00211
(OnPrem)_identityserver.scwdp.zip"
        SqlDbPrefix = $prefix
        SqlServer = $SqlServer
        SqlSecurityUser = $SqlAdminUser
        SqlSecurityPassword = $SqlAdminPassword
        SitecoreIdentityCert = $certParamsForIdentityServer.CertificateName
        Sitename = $IdentityServer
        PasswordRecoveryUrl = $CMurl
        AllowedCorsOrigins = $CMurl
        ClientSecret = $ClientSecret
        LicenseFile = "$PSScriptRoot\license.xml"
    }

Install-SitecoreConfiguration @identityParams -Verbose
```

5. Save the script in the `c:\resourcefiles` folder as `install.ps1`.
6. In a PowerShell command line, navigate to the `c:\resourcefiles` folder and run the following command:

```
.\install.ps1
```

7. When Identity Server installation is completed, on the Identity Server instance, open the `Config\production\Sitecore.IdentityServer.Host.xml` file and modify the connection string for the *Core* database:

   ```
   <ConnectionString>Data Source=.;Initial Catalog=Sitecore.Core;User
   ID=coreuser;Password=Test12345</ConnectionString>
   ```

   If you store users separately, you should edit the connection string for the *security* database.

8. The `PasswordRecoveryUrl` and the `AllowedCorsOriginsGroup1` settings should point to a valid URL for a Sitecore instance:

   ```
   <PasswordRecoveryUrl>http://sitecore_instance/sitecore/login?rc=1</PasswordRecoveryUrl>
   <AllowedCorsOriginsGroup1>http://sitecore_instance</AllowedCorsOriginsGroup1>
   ```

   Note
   In a scaled environment, these settings should point to the Content Management server.

9.1.1 Configuring Sitecore Identity Client

Every Sitecore instance is a default client for Sitecore Identity Server.

After installing Sitecore Identity Client, you must configure two Sitecore Identity Client parameters:

1. To register the Sitecore Identity server on a Sitecore instance, in the `App_Config\Sitecore\Owin.Authentication.IdentityServer\Sitecore.Owin.Authentication.IdentityServer.config` file, add the Identity server URL to the `identityServerAuthority` variable:

   ```
       <sc.variable name="identityServerAuthority"
   value="https://SitecoreIdentityServerHost" />
   ```

2. On the Sitecore instance, in the `ConnectionStrings.config` file, add the `sitecoreidentity.secret` connection string and enter the value for the *ClientSecret* variable that you used when you deployed the Sitecore Identity server:

   ```
   <add name="sitecoreidentity.secret" connectionString="Random_STRING" />
   ```

9.2 Upgrading Sitecore Identity Server

If you have upgraded from Sitecore 9.1.0 or later, you must upgrade Sitecore Identity Server.

Before you upgrade to a new version of the Sitecore Identity Service, create a backup of the existing Sitecore Identity Server directory.

To upgrade the Sitecore Identity Server:

1. On the Sitecore Identity Server instance, stop the IIS application pool.
2. Extract the content of the `Sitecore.IdentityServer 3.0.0 rev. 00211 (OnPrem)_identityserver.scwdp.zip` file to the existing Sitecore Identity Server directory and replace all the files except:
   - `Config\production\Sitecore.IdentityServer.Host.xml`
   - `sitecoreruntime\license.xml`
3. Delete the following files from the root folder:
   - `Sitecore.IdentityServer.Host.pdb`
   - `Sitecore.Plugin.IdentityProvider.AzureAd.pdb`
   - `Sitecore.Plugin.IdentityServer.pdb`
   - `Sitecore.Plugin.IdentityProviders.pdb`
   - `Sitecore.Configuration.Factory.pdb`
4. Start the IIS application pool again.

Chapter 10

Upgrade Multiple Instances

The upgrade procedure for multiple instances depends on your environment architecture.

For a single instance, you must follow all the steps in this guide once.

In a scaled environment, you must repeat the steps in this guide for each instance, with the exception of actions related to the database, which you must follow only once for each database .

This chapter contains the following section:

10.1 Upgrade a Scaled Environment With Multiple Instances

To upgrade a scaled environment with multiple instances, for example, a scaled environment that consists of Content Delivery, Content Management, Processing and Reporting instances that are connected to the same databases – *Core*, *Master*, *Web*, and *Reporting*, you must update each instance.

Content Management Servers

Complete all the steps in this guide on *one* of Content Management servers.

Note
The database upgrade steps *must* be executed *once* for each copy of each database.

Content Management, Delivery, Processing, and Reporting Servers

When one of the Content Management servers has been updated, you must:

1. Synchronize the files between this server and all the other Content Management, Delivery, Processing, and Reporting servers.
2. On each server, in the `Web.config` file, in the `Role:define`, `Search:define` and `Security:define` settings, set the appropriate values.
3. Update the `ConnectionStrings.config` file.

Scaled XConnect Environment

If you are running Sitecore XP 9.0.2 with the scaled XP1 XConnect topology, you must perform all the upgrade steps described in the chapter Upgrade Sitecore xConnect in alignment with the instructions in the Sitecore scaling scenarios documentation.

The Sitecore scaling documentation notes the path to the Sitecore Marketing Automation Engine service and the Sitecore XConnect Search Indexer service.

Chapter 11

Appendix A: The xConnect Application Preparation Script

You can use following Windows batch script to prepare the xConnect application for upgrade. The script deletes files and folders according to the instructions in the section *Update the Files*.

To use the script:

1. Create a text file with the following code and save it with a `.bat` file extension.

```
:: Delete files but keep the folders
del bin /q /s

del App_Data\collectiondeployment /q /s
for /d %%d in ("App_Data\Config\*") do rmdir /s /q %%d
del App_Data\Diagnostics /q /s
del App_Data\Models /q /s
del App_Data\solrcommands /q /s

del App_Data\jobs\continuous\AutomationEngine /q

rmdir App_Data\jobs\continuous\AutomationEngine\App_Data\Config\sitecore /q /s
del App_Data\jobs\continuous\AutomationEngine\App_Data\Diagnostics /q /s

del App_Data\jobs\continuous\IndexWorker /q

rmdir App_Data\jobs\continuous\IndexWorker\App_Data\Config\sitecore /q /s
del App_Data\jobs\continuous\IndexWorker\App_Data\Diagnostics /q /s
del App_Data\jobs\continuous\IndexWorker\App_Data\Models /q /s

del App_Data\jobs\continuous\ProcessingEngine /q

rmdir App_Data\jobs\continuous\ProcessingEngine\App_Data\Config\sitecore /q /s
del App_Data\jobs\continuous\ProcessingEngine\App_Data\Diagnostics /q /s
del App_Data\jobs\continuous\ProcessingEngine\App_Data\Models /q /s

:: Delete language folders from IndexWorker if they exist
rmdir App_Data\jobs\continuous\IndexWorker\de-de /q /s
rmdir App_Data\jobs\continuous\IndexWorker\es-es /q /s
rmdir App_Data\jobs\continuous\IndexWorker\fr-fr /q /s
rmdir App_Data\jobs\continuous\IndexWorker\it-it /q /s
rmdir App_Data\jobs\continuous\IndexWorker\ja-jp /q /s
rmdir App_Data\jobs\continuous\IndexWorker\ko-kr /q /s
rmdir App_Data\jobs\continuous\IndexWorker\pt-br /q /s
rmdir App_Data\jobs\continuous\IndexWorker\ru-ru /q /s
rmdir App_Data\jobs\continuous\IndexWorker\zh-cn /q /s
rmdir App_Data\jobs\continuous\IndexWorker\zh-tw /q /s

:: Delete language folders if they exist
for /d %%d in ("bin\*") do rmdir /s /q %%d
```

2. Copy the file to the root folder of the xConnect application.

3. In a command line, execute the `.bat` file from the xConnect root folder.