Sitecore Experience Platform Upgrade Guide

Sitecore Experience Platform 9.2.0

Upgrade to Sitecore XP 9.2.0 from Sitecore XP 8.1.0 - 9.0.1

Table of Contents

Chapter 1

Introduction

This guide describes how to upgrade from Sitecore Experience Platform 8.1.0 – 9.0.1 to Sitecore Experience Platform 9.2.0 and applies only to the on-premise version.

If you are upgrading from Sitecore XP 9.0.2 or later, use the *Upgrade to Sitecore XP 9.2.0 from Sitecore XP 9.0.2 or later guide* available at the Sitecore Downloads site.

If you are upgrading from Sitecore XP 8.1.0 – 9.0.1, you can upgrade directly to Sitecore XP 9.2.0. However, if your solution is based on an earlier version of Sitecore, you must first upgrade to Sitecore XP 8.1.0 before you can upgrade to Sitecore XP 9.2.0.

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 not all Sitecore functionality will be available during the process.

Chapter 2

Prerequisites

This chapter describes the prerequisites required to upgrade to Sitecore XP 9.2.0.

This chapter contains the following sections:

- Requirements
- Files and scripts for upgrading

2.1 Requirements

Before you upgrade to Sitcore XP 9.2.0, you should 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.

2.1.1 Microsoft Visual C++ 2015 Redistributable Requirements

Sitecore XP 9.0.1 introduced a new prerequisite for the Microsoft Visual C++ 2015 Redistributable. For more information, see https://www.microsoft.com/en-us/download/details.aspx?id=53587.

Note
This redistributable might already be installed with Microsoft Windows. Without it, Sitecore XP will fail to start up with the message: *Could not load file or assembly 'ChilkatDotNet46.dll' or one of its dependencies. The specified module could not be found.*

2.2 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 8.1 – 9.0.1, you must have:

- One of the following files depending on which version of Sitecore XP you are upgrading from:
  - `Sitecore 9.2.0 rev. 002893 (update package for 8.1).zip`
  - `Sitecore 9.2.0 rev. 002893 (update package for 8.2).zip`
  - `Sitecore 9.2.0 rev. 002893 (update package for 9.x).zip`

  Each of these files contains the appropriate update package for that version of Sitecore, for example:

  - `Sitecore 9.2.0 rev. 002893 (update package for 9.x).update`

  Unpack the appropriate version and save it in a suitable location.
- `Sitecore Update Installation Wizard 4.0.1 rev.00153 for 8.x-9.0.zip`
- `Sitecore 9.2.0. rev. 002893 (upgrade files).zip`

  This file is noted as **Configuration files for upgrade** on the Download site and contains:

  - `Database Upgrade Script.zip`

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

    - `CMS_core.sql`
    - `CMS_core_master_web.sql`
    - `SXP_processing_tasks.sql`
    - `SXP_reporting.sql`
    - `SXP_reporting_migrate_experience_analytics_data.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 every version of Sitecore 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.
  - `Disable xDB.zip`

    This file contains:

    - `Disable.xDB.config`
    - `Disable.xAnalytics.config`
  - `MarketingDefinitionsUpgrade.zip`

    This file contains:

    - `RestoreDeletedMarketingTaxonomies.aspx`
    - `RestoreDeletedMarketingDefinitions.aspx`

        - `DefinitionItemsNameValidator.aspx`
    - `Databases.zip`

      This file contains:

        - `Sitecore.Analytics.dacpac`
        - `Sitecore.Exm.master.dacpac`
        - `Sitecore.Experienceforms.dacpac`
        - `Sitecore.Messaging.dacpac`
        - `Sitecore.Processing.tasks.dacpac`
- xDB Data Migration Tool – required if you have existing analytics data in MongoDB that you want to use with Sitecore XP 9.2.0.

Note
Upgrading a solution that uses Oracle databases is *not* supported for any version of Sitecore XP 8.1 or later.

Chapter 3

Prepare for the Upgrade

Before you upgrade to Sitecore XP 9.2.0, you must complete preparatory tasks.

This chapter describes the preparatory tasks for each upgrade path and includes a section describing the tasks that you must perform for all the upgrade paths.

For more information about breaking changes and new features, see the release notes.

This chapter contains the following sections:

3.1 Disable the Sitecore Modules

You must disable any Sitecore modules that you are using.

After the Sitecore Experience Platform is upgraded, you must upgrade all the individual modules.

3.1.1 Web Forms for Marketers

SitecoreXP 9.2.0 does not support the Web Forms for Marketers module. This module was deprecated with the release of Sitecore XP 9.1.

If you are using the Web Forms for Marketers module, you must disable it before you upgrade to Sitecore XP 9.2.0.

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`

3.1.2 Email Experience Manager

If you upgrade from Sitecore XP 8.x and used Email Experience Manager module, you should disable it before upgrade:

1. Delete the following folders:
   - `\App_Config\Include\EmailExperience`
   - `\App_Config\Include\Z.EmailExperience`
2. Delete the following file:
   - `\App_Config\Include\ExperienceProfile\Sitecore.ExperienceProfile.Reporting.Sitecore.EmailExperience.ExperienceProfile.config`
3. In the `\bin` folder of your website, delete all the files that begin with `Sitecore.EmailCampaign`, `Sitecore.EDS`, or `Sitecore.ExM`.

3.1.3 Sitecore Experience Accelerator

If you use Sitecore Experience Accelerator and want to upgrade it, follow the instructions in the SXA Upgrade Guide.

3.1.4 Data Migration Tool

If you used the Data Migration Tool 2.0 to migrate your data to Sitecore XP 9.0, you must uninstall the tool before upgrading to Sitecore XP 9.2.0. After you have migrated your data, you no longer need the tool.

For more information about how to uninstall the Data Migration Tool, see the Data Migration Tool Installation Guide.

If for some reason you cannot uninstall the tool, you must remove the `Sitecore.DataExchange.Tools.XdbDataMigration.dll` from the \`Website\bin` folder.

3.1.5 Disable active Sitecore services

- Delete all `*.JavaScriptServices.*` files from the `bin` and `App_config` folders (and any child folders) in the root `Website` folder.
- Remove `Sitecore.LayoutService.Jss.config` from `App_Config\Sitecore\LayoutService\`
- Remove `zSitecore.JavaScriptServices.GraphQL.config` from `App_Config\Sitecore\Services.GraphQL\`

3.1.6 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.

3.2 To Upgrade from Sitecore XP 8.1 – Sitecore XP 8.2 Update 7

Important
If you are upgrading from Sitecore XP 8.1.0 – Sitecore XP 8.2 Update 7 to Sitecore XP 9.2.0, you must perform all the steps described in this section.

If you are upgrading from Sitecore XP 9.0.0 – 9.0.1 to Sitecore XP 9.2.0, you do not have to perform all the steps described in this section.

3.2.1 Restore Deleted Marketing Taxonomies and Marketing Definitions

If you are using the Sitecore Experience Database (xDB) functionality, (the `Xdb.enabled` setting is set to `true`, and xDB is enabled in the license), before you install the upgrade package, you must restore any of the standard marketing taxonomies and marketing definitions that you have deleted from your Sitecore installation.

When the upgrade is completed, you can delete these unwanted marketing taxonomies and marketing definitions again. This process is described later in this document.

Before you restore anything:

1. Unpack the `MarketingDefinitionsUpgrade.zip` file.
2. Copy the following pages to the `\sitecore\admin` folder of your website:
   - `RestoreDeletedMarketingTaxonomies.aspx`
   - `RestoreDeletedMarketingDefinitions.aspx`

To restore deleted marketing taxonomy items:

1. Open the *Restore Deleted Marketing Taxonomies* page by entering the following URL in your web browser:

   http://<hostname>/sitecore/admin/RestoreDeletedMarketingTaxonomies.aspx

   Important
   This page requires direct access to the *Reporting* database. Ensure that the current Sitecore instance has the appropriate connection string in the `\App_Config\ConnectionStrings.config` file and that the database is accessible.

2. Click *Restore deleted taxonomy items*.

   All the default marketing taxonomy items are restored to the *Master* database. A file called `RestoredTaxonomies_{timestamp}.dat` is created in the `\App_Data` folder. This file lists all of the restored items. When the upgrade is completed, if you want to delete these unwanted marketing taxonomies again, you can use this file to identify them.

To restore deleted marketing definition items:

1. To open the *Restore Deleted Marketing Definitions* page, enter the following URL in your web browser:

   http://<hostname>/sitecore/admin/RestoreDeletedMarketingDefinitions.aspx

> **Important**
> This page uses the Marketing Operations repository in the *Reporting* database. The current Sitecore instance must therefore have access to the *Reporting* database.

2. Click **Restore deleted definition items**.

   All the default marketing definition items are restored from the *Reporting* database to the *Master* database. A file called `RestoredDefinitions_{timestamp}.dat` is created in the `\App_Data` folder. This file lists all of the restored items. When the upgrade is completed, if you want to remove these marketing definitions again, you can use this file to identify them.

3.2.2 Validate the Names of the Marketing Definition Items

If you are using the Sitecore Experience Database (xDB) functionality or if you are using the tracking functionality (the `Xdb.Tracking.Enabled` setting is set to `true`), you must validate the names of the marketing definition items and address any errors.

To validate the names of the marketing definition items:

1. Unpack the `MarketingDefinitionsUpgrade.zip` file.
2. Copy the `DefinitionItemsNameValidator.aspx` page to the `\sitecore\admin` folder of your website.
3. Open the *Definition Items Name Validator* page by entering the following URL in your web browser:

   http://<hostname>/sitecore/admin/DefinitionItemsNameValidator.aspx
4. Click **Validate**.

   The names of the definition items are validated, and a report called `DefinitionItemsNameValidationErrors_{timestamp}.dat` is saved in the `\App_Data` folder. This report lists all the validation errors that occurred.
5. Ensure you manually fix all the issues that were found, and then validate the names again.

   Repeat this procedure until all the names have been validated.

3.2.3 Upgrade the Databases

To upgrade the databases:

1. Execute the `CMS_core_master_web.sql` script for the *Core*, *Master*, and *Web* databases.
2. Execute the `CMS_core.sql` script for the *Core* database only.
3. If you are using the xDB functionality, you must execute the `SXP_reporting.sql` script for the *Reporting* database.

3.2.4 Deploy the New Databases

In Sitecore XP 9.0, the format of all Sitecore databases changed. Existing databased are upgraded to Data-tier Application Component Packages (DACPAC) format. The databases are upgraded with the scripts in the `Sitecore 9.2.0 rev. 002893 (upgrade files).zip` package which is in the `Databases.zip` package.

To use the Sitecore Forms component and the Sitecore Experience Database (xDB) functionality, you must deploy the new databases.

Sitecore Forms

To use the SitecoreForms component:

1. Deploy the `Sitecore.Experienceforms.dacpac` to SQL Server.
2. In the `\App_Config` folder, in the `ConnectionStrings.config` file, add the connection string for the `Sitecore.ExperienceForms` database. For example:

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

The `Experience Forms` connection string is required for the *content management* and *content delivery* server roles.

xDB

To use the xDB functionality:

1. Deploy the `Sitecore.Processing.tasks.dacpac` to SQL Server.
2. To add a user to the database, run the following script:

```
Use [Sitecore.Processing.Tasks]
Go

alter database [Sitecore.Processing.Tasks]
set containment = partial
go

CREATE USER [user] WITH PASSWORD = 'password';
GO

EXEC sp_addrolemember 'db_datareader', [user];
EXEC sp_addrolemember 'db_datawriter', [user];
GO

GRANT EXECUTE TO [user];
GO
```

3. In the `\App_Config` folder, in the `ConnectionStrings.config` file, add the connection string for the `Xdb.Processing.Tasks` database. For example:

```
<add name="xdb.processing.tasks" connectionString="user id=user;password=password;Data
Source=(server);Database=Sitecore.Processing.Tasks" />
```

   The `xdb.processing.tasks` connection string is only required for the *processing* server role.

3.2.5 Enable Email Experience Manager

Note
You must manually migrate your EXM data to Sitecore XP 9.2 – it is not migrated during the upgrade.

For more information about migrating your EXM data, see the Migration Guide.

To enable EXM:

1. Deploy the following files to SQL server:
   - `Sitecore.Exm.master.dacpac`
   - `Sitecore.Messaging.dacpac`

2. You must create SQL logins with the appropriate access rights to the *Exm.master* and *Messaging* databases. These SQL logins must be members of the *db_datareader* and *db_datawriter* security roles.

   To create these logins, update the highlighted parameters in the following script and run it for both databases.

   For example, for *Messaging* database:

```
:SETVAR DatabasePrefix xp0
:SETVAR UserName messaginguser
:SETVAR Password Test12345
:SETVAR MessagingSuffix _Messaging
GO

IF(SUSER_ID('$(UserName)') IS NULL)
BEGIN
       CREATE LOGIN [$(UserName)] WITH PASSWORD = '$(Password)'; END;
GO

USE [$(DatabasePrefix)$(MessagingSuffix)]
IF NOT EXISTS (SELECT * FROM sys.database_principals WHERE name = N'$(UserName)')
BEGIN
    CREATE USER [$(UserName)] FOR LOGIN [$(UserName)] END;
GO

EXEC sp_addrolemember 'db_datareader', '$(UserName)'
EXEC sp_addrolemember 'db_datawriter', '$(UserName)'
GO
```

3. In the `\App_Config` folder, in the `ConnectionStrings.config` file, add connection strings for the *Exm.master* and the *Messaging* databases.

   For example:

```
<add name="exm.master" connectionString="user id=exmmasteruser;password=Test12345;Data
Source=(server);Database=xp0_Exm.master" />
<add name="messaging" connectionString="user id=messaginguser;password=Test12345;Data
Source=(server);Database=xp0_Messaging" />
```

   The `exm.master` connection string is only required on content management servers.

   The `messaging` connection string is required for every server role.

4. Add the following three connection strings:

   - `EXM.InternalApiKey`

     You only need to add this connection string if you want to add a dedicated email dispatch server.

   - `EXM.CryptographicKey`
   - `EXM.AuthenticationKey`

   The keys must be represented in hexadecimal format by 64 characters and you can only use the symbols 0-9 and A-F.

   For example:

```
<add name="EXM.CryptographicKey" connectionString=
"E040C938FC9E4EBC3E93330B0F7837F284207B8180DB64CB5B6ABEB1AFBF6F5B" />
```

3.3 To Upgrade from Sitecore XP 9.0.0 – 9.0.1

Important
If you are upgrading from Sitecore XP 9.0.0 – 9.0.1 to Sitecore XP 9.2.0 or later and are not using the xDB functionality, you do not have to perform the steps described in this section.

3.3.1 Disable Experience Analytics

To prevent *item:saved* events from occurring during the upgrade process, disable Experience Analytics:

1. Unpack the `Sitecore 9.2.0 rev. 002893 (upgrade files).zip` file and the `Disable xDB.zip` file and locate the `Disable.xAnalytics.config` file.
2. In the `\App_Config\Include` folder, create a `\Z.Custom` folder.
3. Copy the `Disable.xAnalytics.config` file to the `\App_Config\Include\Z.Custom` folder.

Important
When the upgrade is finished, to re-enable the Experience Analytics events, you must remove the `Disable.xAnalytics.config` file.

3.3.2 Stop the Windows Services

You must stop the following Windows services:

- The Sitecore XConnect Search Indexer service.
- The Sitecore Marketing Automation Engine service.

3.3.3 Upgrade the Databases

To upgrade the databases, you must run the following scripts:

1. Execute the `CMS_core.sql` script for the *Core* database.
2. Execute the `SXP_processing_tasks.sql` script for the *Processing.tasks* database.
3. Execute the `SXP_reporting.sql` script for the *Reporting* database.

3.3.4 Enable Email Experience Manager (EXM)

This section contains information about how to upgrade EXM from:

- Sitecore XP 9.0 Initial Release
- Sitecore XP 9.0.1

Upgrade from Sitecore XP 9.0 Initial Release

Note
If you are upgrading from Sitecore XP 9.0.1, skip this section.

To enable EXM:

1. Deploy the following files to SQL Server:

   - `Sitecore.Exm.master.dacpac`
   - `Sitecore.Messaging.dacpac`

2. You must create SQL logins with the appropriate access rights to the *Exm.master* and *Messaging* databases. These SQL logins must be members of the *db_datareader* and *db_datawriter* security roles.

   To create these logins, update the highlighted parameters in the following script and run it for both databases.

   For example, for *Messaging* database:

```
:SETVAR DatabasePrefix xp0
:SETVAR UserName messaginguser
:SETVAR Password Test12345
:SETVAR MessagingSuffix _Messaging
GO

IF(SUSER_ID('$(UserName)') IS NULL)
BEGIN
       CREATE LOGIN [$(UserName)] WITH PASSWORD = '$(Password)'; END;
GO

USE [$(DatabasePrefix)$(MessagingSuffix)]
IF NOT EXISTS (SELECT * FROM sys.database_principals WHERE name = N'$(UserName)')
BEGIN
    CREATE USER [$(UserName)] FOR LOGIN [$(UserName)] END;
GO

EXEC sp_addrolemember 'db_datareader', '$(UserName)'
EXEC sp_addrolemember 'db_datawriter', '$(UserName)'
GO
```

3. In the `\App_Config` folder, in the `ConnectionStrings.config` file, add connection strings for the *Exm.master* and the *Messaging* databases.

   For example:

```
<add name="exm.master" connectionString="user id=exmmasteruser;password=Test12345;Data
Source=(server);Database=xp0_Exm.master" />
<add name="messaging" connectionString="user id=messaginguser;password=Test12345;Data
Source=(server);Database=xp0_Messaging" />
```

   The `exm.master` connection string is only required on content management servers.

   The `messaging` connection string is required for every server role.

4. Add the following three connection strings:

   - `EXM.InternalApiKey`

     You only need to add this connection string if you want to add a dedicated email dispatch server.

   - `EXM.CryptographicKey`
   - `EXM.AuthenticationKey`

   The keys must be represented in hexadecimal format by 64 characters and you can only use the symbols 0-9 and A-F.

   For example:

```
<add name="EXM.CryptographicKey" connectionString=
"E040C938FC9E4EBC3E93330B0F7837F284207B8180DB64CB5B6ABEB1AFBF6F5B" />
```

Upgrade the *EXM.Master* database from Sitecore XP 9.0.1

If you are upgrading from Sitecore XP 9.0.1, exectute the following scripts for the *EXM.Master* database:

- `SXP_exm_master_dispatchqueue.sql`
- `SXP_exm_master_suppressions.sql`
- `SXP_exm_master_update_suppressions.sql`
- `SXP_exm_master_save_suppressions.sql`

3.4 Final preparations for all versions

3.4.1 Add Security and PackageManagementServiceUrl Connection Strings

Before you upgrade to Sitecore XP 9.2.0, you must add *security* and *package management* connection strings.

In Sitecore XP 9.0.2, we added a new *PackageManagementServiceUrl* connection that enables you to use the the Update Center to upgrade your Sitecore installation.

To add the connection strings:

1. In the `\App_Config` folder, open the `ConnectionStrings.config` file.
2. Add the *security* connection string, for example:

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

3. Add the *PackageManagementServiceUrl* connection string:

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

3.4.2 Indexing and Performance

Note
If you are upgrading from Sitecore XP 8.2 Initial release or earlier, skip this section.

Indexing is performed during the upgrade. Depending on the version of Sitecore you are upgrading from, a number of items can be changed by the update, and this can cause the upgrade to take a long time.

To speed up the upgrade process, disable indexing by removing the following processor from the `Sitecore.ContentSearch.config` file:

```
<handler type="Sitecore.ContentSearch.Events.PackagingEventHandler, Sitecore.ContentSearch"
method="OnPackageInstallItemsEndRemoteHandler"/>
```

3.4.3 Stop All Content Tests and Disable xDB

If you are using the Sitecore Experience Database (xDB) functionality, you must perform these steps.

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 have the active content tests stopped by either having the tests select a winner or by manually canceling the tests.

Disable xDB

To disable xDB:

1. Unpack the `Sitecore 9.2.0 rev. 002893 (upgrade files).zip` file and the `Disable xDB.zip` file.

2. Create a `\App_Config\Include\Z.Custom` folder, and copy the `Disable.xDB.config` file to this folder.

   Important
   When the upgrade is finished, you must re-enable xDB by removing the `Disable.xDB.config` file from the `\App_Config\Include\Z.Custom` folder.

Chapter 4

Upgrade xConnect to Sitecore XP 9.2.0

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:

This chapter describes how to upgrade xConnect from:

- Sitecore XP 9.0.0
- Sitecore XP 9.0.1

Sitecore XP 9.1.0 introduced the new Sitecore Cortex™ Processing Engine role. This chapter describes how to install it.

> **Important**
> If you are upgrading from Sitecore XP 9.0.0 – Sitecore XP 9.0.1 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:

- Prepare to upgrade xConnect Server
- Upgrade the Databases
- Update the Files
- Configure the Search Provider
- Configure the Azure Key Vault
- Resume the Windows services

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.

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 database upgrade scripts including:

   - `SXP_marketingautomation.sql`
   - `SXP_referencedata.sql`
   - `SXP_processing_pools.sql`
   - `SXP_collection.sql`
   - `SXP_collection_AE_Part1.sql`
   - `SXP_collection_AE_Part2.sql`
   - `SXP_collection_GrantPermissions.sql`
   - `SXP_processing_engine_tasks.sql`
   - `SXP_processing_tasks.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 9.0.1 or earlier, you cannot pause these tasks before the upgrade and resume them afterwards because the extraction cursors were changed in Sitecore XP 9.1.1.

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

4.2 Upgrade the Databases

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

4.2.1 Deploy the Messaging Database

If you are upgrading from Sitecore XP 9.0 Initial Release or earlier, you must:

1. Deploy the *Messaging* database.
2. In the following files:
   - `\xconnect\App_Config\ConnectionStrings.config`
   - `\xconnect\App_data\jobs\continuous\AutomationEngine\App_Config\ConnectionStrings.config`

   Add the *messaging* connection string, for example:

   ```
   <add name="messaging" connectionString="user id=messaginguser;password=Test12345;data
   source=source; Database=Sitecore.Messaging" />
   ```

For more information about deploying the *Messaging* database, see the section *Enable Email Experience Manager*.

Note
You must add the *messaging* connection string on every xConnect server that performs the *Sitecore Marketing Automation Engine* role and the *Sitecore Cortex™ Processing Engine* role.

4.2.2 Upgrade existing Databases

You must upgrade the databases on solutions that are running on Sitecore XP 9.0.0 - Sitecore XP 9.0.1.

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.

Upgrade the Collection database 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]
```

Upgrade the Collection database on MongoDB

Important
If you are going to move from SQL Server 2016 to MongoDB as part of the upgrade, skip this section.

The installation guide for Sitecore XP 9.1.1 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*.

4.2.3 Upgrade the Messaging database

Important
If you are upgrading from Sitecore XP 9.0.1, you must upgrade the *Messaging* database.

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

4.3 Update the Files

Now you can 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`
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\logs`

4. When prompted, replace any files that have the same name.

4.3.1 Add New Connection Strings

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

If you installed a new instance of Solr instead of updating the existing one to Solr 7.5, you must configure the new Solr cores.

To create new Solr cores, in the Solr admin console:

1. Add 2 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`.

2. 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
```

3. In the following connection string files:
   - `xconnect\App_Config\ConnectionStrings.config`
   - `xconnect\App_data\jobs\continuous\IndexWorker\App_config\ConnectionStrings.config`
4. 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 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 the 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.

4.6 Resume the Windows 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. Rebuild the xConnect indexes.

Chapter 5

Install xConnect

If you want to use the Sitecore Experience Database (xDB) functionality in Sitecore XP 9.2, you must install and configure Sitecore xConnect.

Important
If you are upgrading from Sitecore XP 8.1.0 – Sitecore XP 8.2 Update 7, you must install and configure Sitecore xConnect.

If you are upgrading from Sitecore XP 9.0 – Sitecore XP 9.0.1 with xConnect installed, skip this chapter.

If you are upgrading from Sitecore XP 9.0 – Sitecore XP 9.0.1 *without* xConnect installed and want to use the Sitecore Experience Database (xDB) functionality in Sitecore XP 9.2.0, you must perform the tasks described in this chapter.

This chapter contains the following section:

5.1 Install and Configure Sitecore xConnect

For more detailed information about which topologies support xConnect, see the *Sitecore Experience Platform 9.2.0 Installation Guide*. You can download the latest Sitecore Installation Guide from the Sitecore Downloads Page.

To configure Sitecore xConnect for a single instance Sitecore solution:

1. Install xConnect.
2. For a single instance Sitecore solution that you have upgraded, in the `\App_Config` folder, in the `ConnectionStrings.config` file, add the following connection strings and modify all the parameters:

```
  <add name="xconnect.collection" connectionString="https://Sitecore_xconnect"/>
  <add name="xconnect.collection.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;Fi
ndValue="/>
  <add name="xdb.referencedata" connectionString="user id=user;password=password;Data
Source=(server);Database=Sitecore_ReferenceData"/>
  <add name="xdb.referencedata.client" connectionString="https://Sitecore_xconnect"/>
  <add name="xdb.referencedata.client.certificate"
connectionString="StoreName=My;StoreLocation=
LocalMachine;FindType=FindByThumbprint;FindValue="/>
  <add name="xdb.processing.pools" connectionString="user
id=user;password=password;Data Source=(server);Database=Sitecore_Processing.Pools"/>
  <add name="xdb.marketingautomation" connectionString="user
id=user;password=password;Data
Source=(server);Database=Sitecore_MarketingAutomation"/>
  <add name="xdb.marketingautomation.reporting.client"
connectionString="https://Sitecore_xconnect"/>
  <add name="xdb.marketingautomation.reporting.client.certificate"
connectionString="StoreName=My;StoreLocation=
LocalMachine;FindType=FindByThumbprint;FindValue="/>
  <add name="xdb.marketingautomation.operations.client"
connectionString="https://Sitecore_xconnect"/>
  <add name="xdb.marketingautomation.operations.client.certificate"
 connectionString="StoreName=My;StoreLocation=
LocalMachine;FindType=FindByThumbprint;FindValue="/>
  <add name="messaging" connectionString="user id=user;password=password;Data
Source=(server);Database=Sitecore.Messaging"/>
  <add name="sitecore.reporting.client" connectionString="https://sc910_xconnect" />
  <add name="sitecore.reporting.client.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;Fi
ndValue=C7C27AAE1E197AC0B743BDACF18E21C214C9CAB5" />
```

Note
These connection strings are for a single instance Sitecore solution only. For a scaled environment, see the Sitecore server role configuration reference documentation.

Grant permissions to the xConnect certificates

You must grant read permissions to the xConnect client and server certificates to `ApplicationPoolIdentity` of your Sitecore instances.

To grant the read permissions:

1. To open the **Certificate Management** console, in the Windows command prompt, enter `certlm.msc` and press **Enter**.
2. In the left pane, in the tree, expand the Personal node and select **Certificates**.
3. In the right pane, right-click the relevant xConnect certificate, select **All Tasks**, and then select **Manage Private Keys**.
4. In the **Permissions** dialog box, in the **Security** section, ensure that you grant *read* permissions to the relevant Sitecore instance Application Pool Identity.

For example: `IIS AppPool\<YourSitecoreAppPoolName>`.

5. Restart IIS.

Chapter 6

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 9.0 – Sitecore XP 9.0.1, you must install and configure Sitecore Cortex Processing Engine.

If you are upgrading from Sitecore XP 9.0 – Sitecore XP 9.0.1 without xConnect installed, the Sitecore Cortex Processing engine is installed as part of xConnect and you can skip this chapter.

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 section:

6.1 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**
> Please note that processing.webapi.blob, processing.webapi.table apis are deployed with your xConnect instance and should point to you 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 processing engine storage connection string to 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" />
```

6.2 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 7

Install the Upgrade Package

To install the upgrade package and update your website, you must use the Update Installation Wizard.

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

This chapter contains the following sections:

- Install the Upgrade Package
- The Upgrade Package Analysis

7.1 Install the Upgrade Package

To install the upgrade package:

1. On the **Sitecore Launchpad**, click **Control Panel**, and in the **Administration** section, click **Install a package**.
2. Install the `Sitecore Update Installation Wizard 4.0.1 rev. 00153 for 8.x-9.0.zip` package.

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

3. On the **Sitecore Launchpad**, click **Control Panel**, and in the **Administration** section, click **Install an update**.

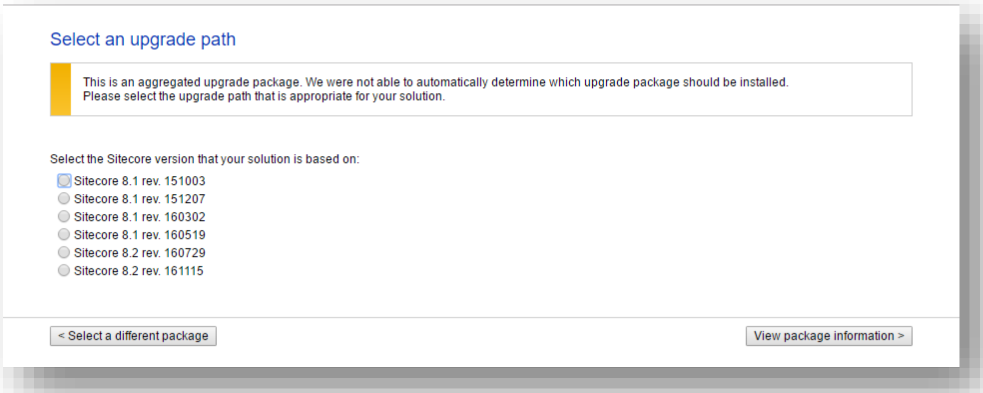
   Alternatively, you can open the **Update Installation Wizard** by entering the following URL in your web browser:

   http://<hostname>/sitecore/admin/UpdateInstallationWizard.aspx

   Note
   If you are running any version of Sitecore XP 8.1, use the URL to open the wizard.

4. On the **Welcome to Sitecore update installation wizard** page, click **Select a package**.
5. On the **Select a package** page, click **Choose File** and navigate to the folder where you saved the update file for Sitecore XP 9.2.0. Select the update file for the Sitecore version that you are upgrading from, and then click **Open**.
6. Click **Package information**.

   If the **Update Installation Wizard** cannot detect your current Sitecore version automatically, on the **Select an upgrade path** page, you must manually select the Sitecore version and revision number that your solution is based on, and then click **View package information**.



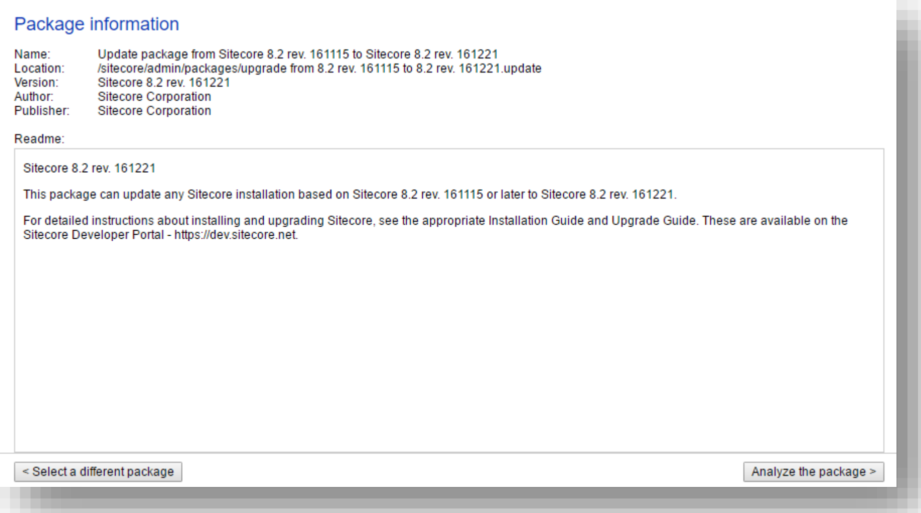
Select an upgrade path

This is an aggregated upgrade package. We were not able to automatically determine which upgrade package should be installed.
Please select the upgrade path that is appropriate for your solution.

Select the Sitecore version that your solution is based on:

○ Sitecore 8.1 rev. 151003
○ Sitecore 8.1 rev. 151207
○ Sitecore 8.1 rev. 160302
○ Sitecore 8.1 rev. 160519
○ Sitecore 8.2 rev. 160729
○ Sitecore 8.2 rev. 161115

7. On the **Package information** page, review the information about the package, and then click **Analyze the package**.



Package information

Name: Update package from Sitecore 8.2 rev. 161115 to Sitecore 8.2 rev. 161221
Location: /sitecore/admin/packages/upgrade from 8.2 rev. 161115 to 8.2 rev. 161221.update
Version: Sitecore 8.2 rev. 161221
Author: Sitecore Corporation
Publisher: Sitecore Corporation

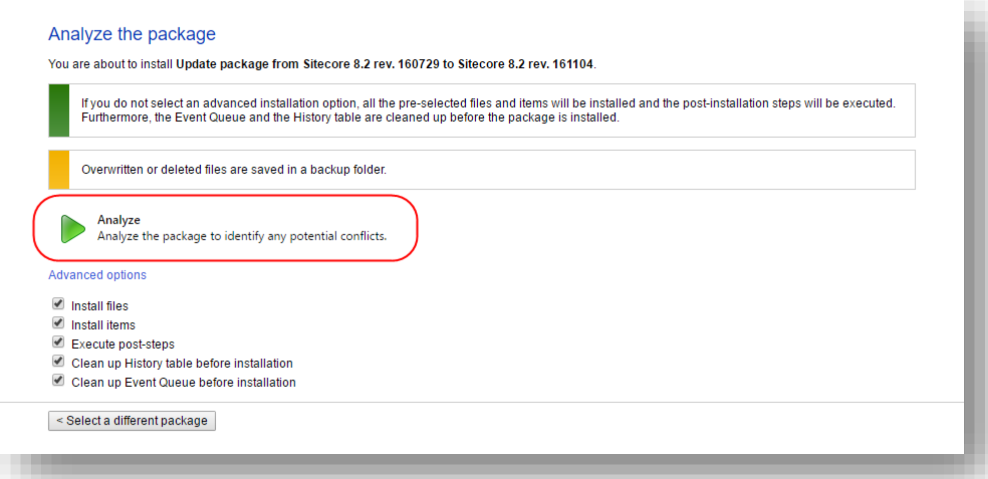
Readme:

Sitecore 8.2 rev. 161221

This package can update any Sitecore installation based on Sitecore 8.2 rev. 161115 or later to Sitecore 8.2 rev. 161221.

For detailed instructions about installing and upgrading Sitecore, see the appropriate Installation Guide and Upgrade Guide. These are available on the Sitecore Developer Portal - https://dev.sitecore.net.

8. On the **Analyze the package** page, click **Advanced options**, review the installation options, and then click **Analyze**.



Analyze the package

You are about to install **Update package from Sitecore 8.2 rev. 160729 to Sitecore 8.2 rev. 161104**.

If you do not select an advanced installation option, all the pre-selected files and items will be installed and the post-installation steps will be executed. Furthermore, the Event Queue and the History table are cleaned up before the package is installed.

Overwritten or deleted files are saved in a backup folder.

Analyze
Analyze the package to identify any potential conflicts.

Advanced options

☑ Install files
☑ Install items
☑ Execute post-steps
☑ Clean up History table before installation
☑ Clean up Event Queue before installation

In the **Advanced options** section, you must select the **Execute post-steps** check box, regardless of the other check boxes you may have selected.

The post-steps are performed at the end of the upgrade process. These steps can include items or file operations that are executed by the Sitecore API, and can vary according to the Sitecore version that you are upgrading from.

During the analysis, the installation wizard identifies any potential conflicts in the configuration files being upgraded and any breaking changes in custom code.

9. After the analysis process is complete, click **Analysis result**.

10. If the **Update Installation Wizard** does not find any conflicts, click **Install the package**, and skip to the section *Configure Sitecore*.

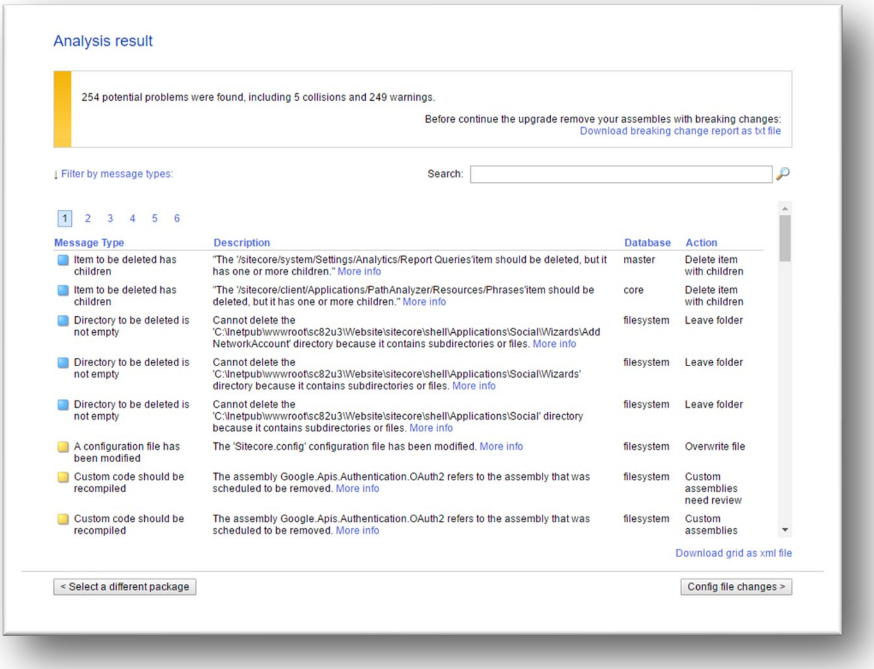
7.2 The Upgrade Package Analysis

The analysis phase detects any problems that can occur during the upgrade. The analysis report contains details about collisions, errors, and warnings that you must deal with before performing the upgrade.

Collisions, errors, and warnings can occur when there is a risk of losing data. For example, you can lose data when the upgrade involves deleting an item that has children or a folder that contains files.

In the analysis report, collisions are marked with a blue square 🟦 and warnings with a yellow square 🟨. However, if the wizard finds errors, it will display a red square 🟥.

The action that the wizard will take to resolve each collision or warning is listed in the **Action** column.



Analysis result

254 potential problems were found, including 5 collisions and 249 warnings.

Before continue the upgrade remove your assembles with breaking changes:
Download breaking change report as txt file

Message Type	Description	Database	Action
🟦 Item to be deleted has children	"The '/sitecore/system/Settings/Analytics/Report Queries'item should be deleted, but it has one or more children." More info	master	Delete item with children
🟦 Item to be deleted has children	"The '/sitecore/client/Applications/PathAnalyzer/Resources/Phrases'item should be deleted, but it has one or more children." More info	core	Delete item with children
🟦 Directory to be deleted is not empty	Cannot delete the 'C:\Inetpub\wwwroot\sc82u3\Website\sitecore\shell\Applications\Social\Wizards\Add NetworkAccount' directory because it contains subdirectories or files. More info	filesystem	Leave folder
🟦 Directory to be deleted is not empty	Cannot delete the 'C:\Inetpub\wwwroot\sc82u3\Website\sitecore\shell\Applications\Social\Wizards' directory because it contains subdirectories or files. More info	filesystem	Leave folder
🟦 Directory to be deleted is not empty	Cannot delete the 'C:\Inetpub\wwwroot\sc82u3\Website\sitecore\shell\Applications\Social' directory because it contains subdirectories or files. More info	filesystem	Leave folder
🟨 A configuration file has been modified	The 'Sitecore.config' configuration file has been modified. More info	filesystem	Overwrite file
🟨 Custom code should be recompiled	The assembly Google.Apis.Authentication.OAuth2 refers to the assembly that was scheduled to be removed. More info	filesystem	Custom assemblies need review
🟨 Custom code should be recompiled	The assembly Google.Apis.Authentication.OAuth2 refers to the assembly that was scheduled to be removed. More info	filesystem	Custom assemblies

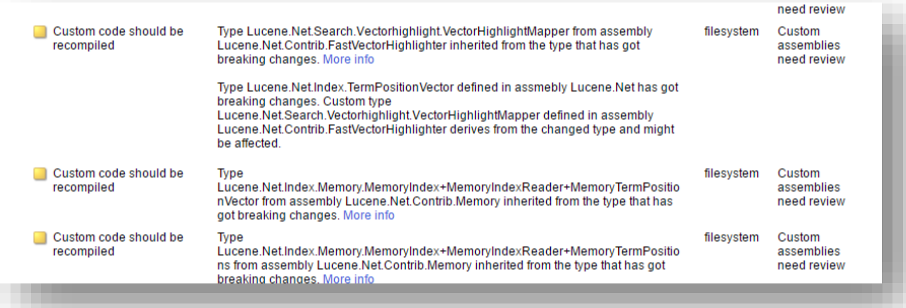
We recommend that you review all the collisions, errors, and warnings, as well as the actions that will be performed to resolve them.

Important
While the majority of the actions ensure the best upgrade experience, if the analysis identifies custom code that must be recompiled or configuration file conflicts that must be resolved, you must address these manually.

7.2.1 Resolve Breaking Changes in Custom Code

If the wizard detects some custom code that will be affected by breaking changes that were introduced in a Sitecore version, the **Analysis result** page displays a warning.

The warning comes with an action and a description. To see more information, click **More info**.



			need review
🟨 Custom code should be recompiled	Type Lucene.Net.Search.Vectorhighlight.VectorHighlightMapper from assembly Lucene.Net.Contrib.FastVectorHighlighter inherited from the type that has got breaking changes. More info Type Lucene.Net.Index.TermPositionVector defined in assmebly Lucene.Net has got breaking changes. Custom type Lucene.Net.Search.Vectorhighlight.VectorHighlightMapper defined in assembly Lucene.Net.Contrib.FastVectorHighlighter derives from the changed type and might be affected.	filesystem	Custom assemblies need review
🟨 Custom code should be recompiled	Type Lucene.Net.Index.Memory.MemoryIndex+MemoryIndexReader+MemoryTermPositionVector from assembly Lucene.Net.Contrib.Memory inherited from the type that has got breaking changes. More info	filesystem	Custom assemblies need review
🟨 Custom code should be recompiled	Type Lucene.Net.Index.Memory.MemoryIndex+MemoryIndexReader+MemoryTermPositions from assembly Lucene.Net.Contrib.Memory inherited from the type that has got breaking changes. More info	filesystem	Custom assemblies need review

The wizard only analyzes `.dll` files that are stored in the `\bin` folder.

To resolve these warnings:

- Remove all the affected `.dll` files from the `\bin` folder.

  If any of your Sitecore configuration files contain references to the `.dll` files that you removed, you must remove these references as well.

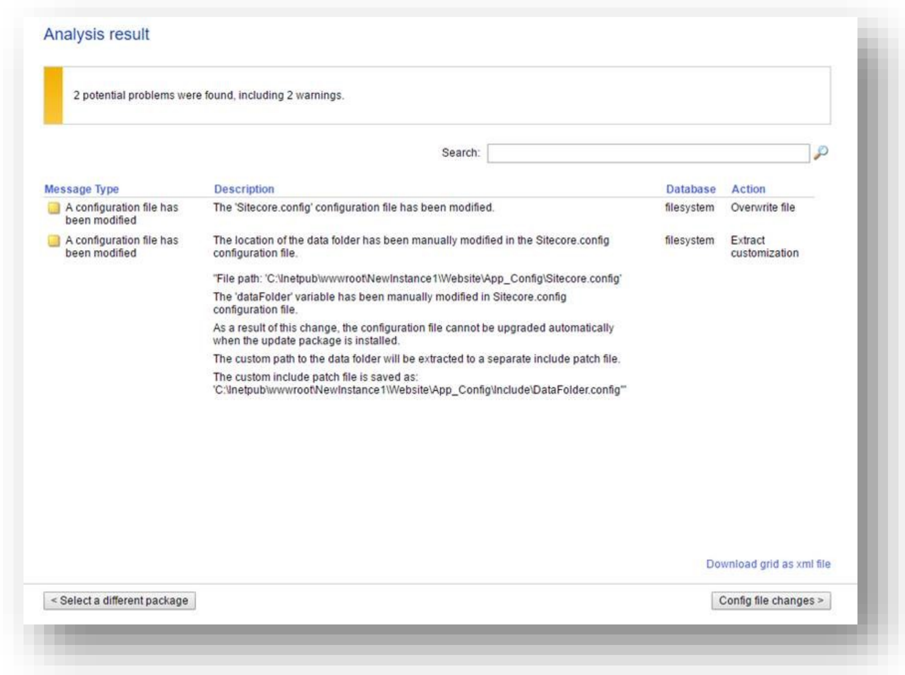
  Or

- If you are sure that the affected code is not used during Sitecore startup or in important Sitecore operations such as item management, security management, file management, or in regular website requests that require Sitecore functionality, you can ignore the warnings.

Note
After you remove the custom code, you must analyze the package again to ensure that the **Analysis result** page contains no errors.

7.2.2 Resolve Configuration File Conflicts

If any Sitecore configuration files have been modified, the **Analysis result** page displays a warning.



Analysis result

2 potential problems were found, including 2 warnings.

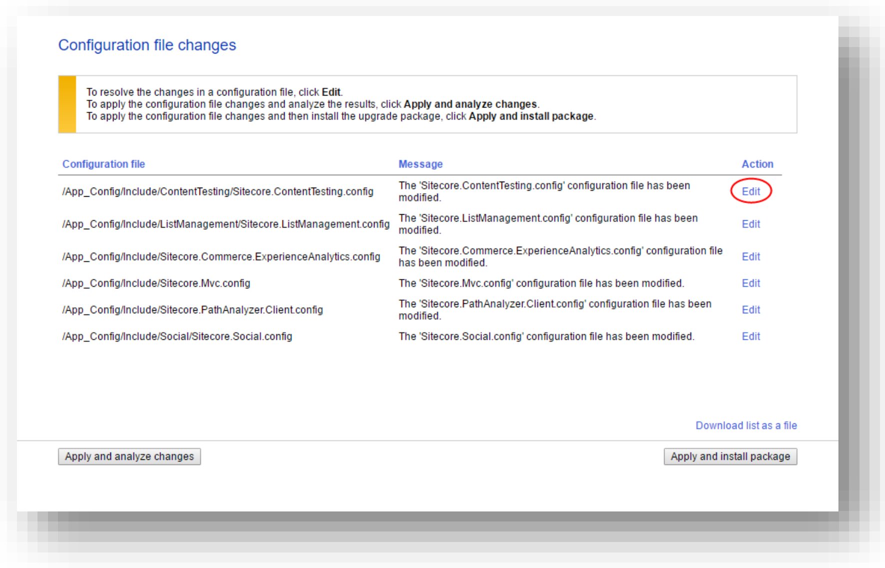
Message Type	Description	Database	Action
A configuration file has been modified	The 'Sitecore.config' configuration file has been modified.	filesystem	Overwrite file
A configuration file has been modified	The location of the data folder has been manually modified in the Sitecore.config configuration file. "File path: 'C:\Inetpub\wwwroot\NewInstance1\Website\App_Config\Sitecore.config' The 'dataFolder' variable has been manually modified in Sitecore.config configuration file. As a result of this change, the configuration file cannot be upgraded automatically when the update package is installed. The custom path to the data folder will be extracted to a separate include patch file. The custom include patch file is saved as: 'C:\Inetpub\wwwroot\NewInstance1\Website\App_Config\Include\DataFolder.config'"	filesystem	Extract customization

Some of these configuration file conflicts can be resolved automatically and others must be resolved manually.

To see more information about a configuration file change, click the description.

To resolve configuration file conflicts:

1. On the **Analysis result** page, click **Config file changes**.



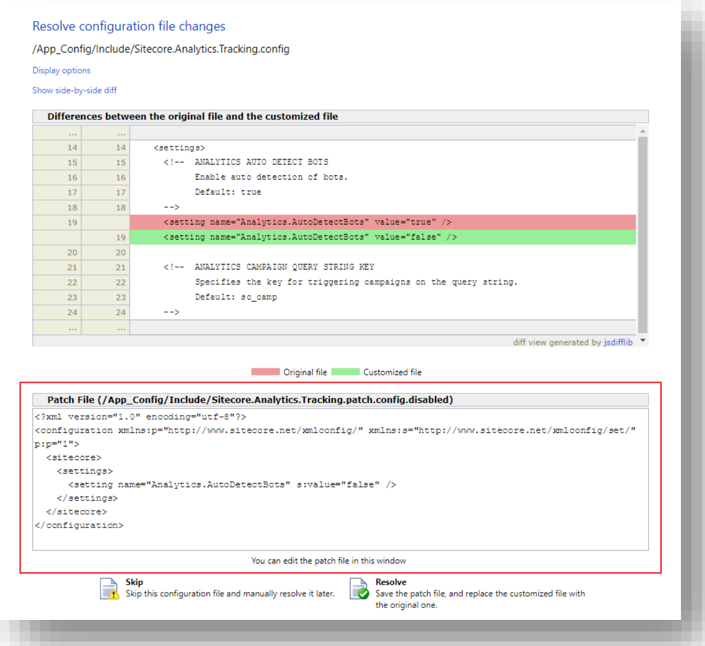
Configuration file changes

To resolve the changes in a configuration file, click **Edit**.
To apply the configuration file changes and analyze the results, click **Apply and analyze changes**.
To apply the configuration file changes and then install the upgrade package, click **Apply and install package**.

Configuration file	Message	Action
/App_Config/Include/ContentTesting/Sitecore.ContentTesting.config	The 'Sitecore.ContentTesting.config' configuration file has been modified.	Edit
/App_Config/Include/ListManagement/Sitecore.ListManagement.config	The 'Sitecore.ListManagement.config' configuration file has been modified.	Edit
/App_Config/Include/Sitecore.Commerce.ExperienceAnalytics.config	The 'Sitecore.Commerce.ExperienceAnalytics.config' configuration file has been modified.	Edit
/App_Config/Include/Sitecore.Mvc.config	The 'Sitecore.Mvc.config' configuration file has been modified.	Edit
/App_Config/Include/Sitecore.PathAnalyzer.Client.config	The 'Sitecore.PathAnalyzer.Client.config' configuration file has been modified.	Edit
/App_Config/Include/Social/Sitecore.Social.config	The 'Sitecore.Social.config' configuration file has been modified.	Edit

2. To automatically resolve all the configuration file conflicts and install the package, click **Apply and install package**. When you are finished, skip to step 5.

3. To manually resolve a configuration file conflict, in the **Action** column for the configuration file, click **Edit**.
4. On the **Resolve configuration file changes** page, review the differences between the original file and the customized file.



Resolve configuration file changes

/App_Config/Include/Sitecore.Analytics.Tracking.config

Display options

Show side-by-side diff

Differences between the original file and the customized file

```
 ...   ...
  14    14      <settings>
  15    15        <!--  ANALYTICS AUTO DETECT BOTS
  16    16              Enable auto detection of bots.
  17    17              Default: true
  18    18        -->
  19              <setting name="Analytics.AutoDetectBots" value="true" />
        19        <setting name="Analytics.AutoDetectBots" value="false" />
  20    20
  21    21        <!--  ANALYTICS CAMPAIGN QUERY STRING KEY
  22    22              Specifies the key for triggering campaigns on the query string.
  23    23              Default: sc_camp
  24    24        -->
 ...   ...
```

diff view generated by jsdifflib

🟥 Original file 🟩 Customized file

Patch File (/App_Config/Include/Sitecore.Analytics.Tracking.patch.config.disabled)

```
<?xml version="1.0" encoding="utf-8"?>
<configuration xmlns:p="http://www.sitecore.net/xmlconfig/" xmlns:s="http://www.sitecore.net/xmlconfig/set/"
p:p="1">
  <sitecore>
    <settings>
      <setting name="Analytics.AutoDetectBots" s:value="false" />
    </settings>
  </sitecore>
</configuration>
```

You can edit the patch file in this window

Skip
Skip this configuration file and manually resolve it later.

Resolve
Save the patch file, and replace the customized file with the original one.

5. To manually resolve the conflicts, in the **Patch File** panel, make the necessary changes, and then click **Resolve**.
6. To skip a conflict and manually resolve it later, click **Skip**.

   You must manually review your configuration patch files after the upgrade is complete.

   Note
   When you skip the file, the **Update Installation Wizard** automatically detects any customizations in the Sitecore configuration files, creates the patch, and replaces the customized file with the new file when it installs the upgrade package.

7. After you have dealt with all the configuration file conflicts, on the **Configuration file changes** page, click **Apply and analyze changes**.

   At this point, the **Update Installation Wizard** restarts the analysis and ensures that all the configuration file conflicts have either been resolved or skipped.

   The **Update Installation Wizard** then creates new patch files with the customizations and disables them by adding the `.disabled` extension to the files.

   Note
   In certain scenarios, the generated patch files can contain customizations that are critical for the Sitecore instance to run correctly. To avoid breaking changes, the **Update Installation Wizard** displays an error on the **Apply and analyze changes** page.

   If this happens, you must enable the patch files by removing the `.disabled` extension, and then refresh the **Update Installation Wizard** page in your browser.

The patch files are created and stored in the same folder as the original configuration files.

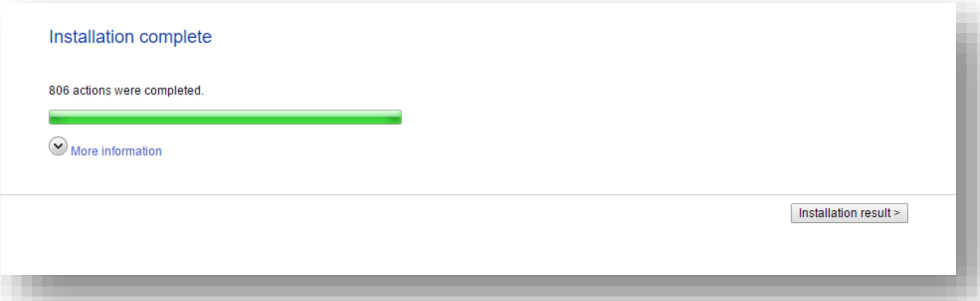
8. When the analysis is complete, click **Analysis result**.

   Note
   If the analysis fails, you must restart the **Update Installation Wizard**. If this does not solve the problem, contact Sitecore support.

9. If the **Update Installation Wizard** does not find any more configuration file conflicts, click **Install the package**.

   Any conflicts that were skipped or left unresolved during the configuration are now resolved automatically.

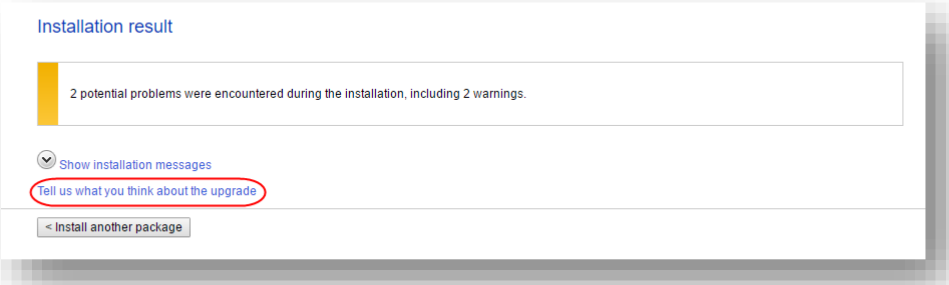
10. When the update process is finished, on the **Installation complete** page, click **Installation result**.



Installation complete

806 actions were completed.

The **Installation result** page contains a report that describes any potential problems that arose as a result of the upgrade process and which require review. The upgrade process performs actions to resolve conflicts automatically and complete the upgrade. However, you must review some of these actions to ensure that there was no unexpected data loss. Usually, the problems are inconsequential and the upgrade is successful.



Installation result

2 potential problems were encountered during the installation, including 2 warnings.

Show installation messages
Tell us what you think about the upgrade
< Install another package

If any errors occurred during the upgrade, you must analyze and resolve them. After you have addressed these errors, you must restart the upgrade either with the default options or, depending on the nature of the problems that you addressed, with the items-only or files-only option selected.

Feedback

When the installation is complete, you can give feedback to the Sitecore development team about your upgrade experience. To do this, on the **Installation result** page, click **Tell us what**

you think about the upgrade. The feedback is anonymous and the feedback form does not collect any instance or license-specific information.

Chapter 8

Configure Sitecore

After you have installed the 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:

8.1 Review Changes in Configuration Files

8.1.1 The Helix Folder Structure

Sitecore XP 9.1.0 or later creates the default Helix folder structure by default. 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.

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

8.1.2 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
It is best practice to store all the generated patch files in the `\Website\App_Config\Include\Z.Custom` folder.

8.1.3 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.

8.2 Specify the Server Role and the Search Provider

Note
If you are upgrading Sitecore XP 9.0.0 – 9.0.1, you do not need to specify the server role or the search provider and can skip this section.

Sitecore XP 9.0.0 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.

All the Sitecore configuration files are enabled by default. The server role and the search provider are controlled by settings in the `Web.config` file.

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

  The default value is *Standalone*, which 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.

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. Consider no longer using Lucene.)
  - Solr
  - Azure

    The default value is *Solr*.

    The Azure Search provider is only supported for Azure Cloud PaaS deployments.

For more information about rule-based configuration, see the Sitecore Documentation.

8.3 Remove Deprecated Indexes

Important
If you are upgrading from Sitecore XP 8.1 – Sitecore 8.2 Update 7 to Sitecore XP 9.2.0, you must perform this step.

If you are upgrading from Sitecore XP 9.0.0 – 9.0.1 to Sitecore XP 9.2.0, skip this step.

You must remove the following deprecated indexes from your search provider:

- `social_messages_master`
- `social_messages_web`
- `sitecore_list_index`
- `sitecore_analytics_index`

To remove the indexes in Lucene:

1. Navigate to `C:\inetpub\wwwroot\<SitecoreInstanceName>\Data\indexes`
2. Delete any folder named after any of the deprecated search indexes.

To remove the indexes in Solr:

1. Navigate to the path where your Solr server is installed, for example: `C:\solr-6.6.2\server\solr`
2. Delete any folder named after any of the deprecated search indexes.

To remove indexes in Microsoft Azure:

1. Log in to the Azure portal.
2. Open the Azure Search service.
3. Navigate to the search index, select the relevant indexes and then click **Delete**.

8.4 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.

8.4.1 Creating 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.

   For example, `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 the following information:

```
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 should update the corresponding setting in the config file.

11. Repeat all the previous steps for each index.

8.4.2 Configure Managed Schema

If you created new cores for indexes, skip this section.

Since Sitecore XP 9.0, Sitecore has supported the Managed Schema approach. Therefore, you no longer need to set up Solr Server to run the Classic Schema.

Important
If you are upgrading from Sitecore XP 9.0.0 – 9.0.1, skip this step.

To update Solr cores to use managed schema:

1. Open the managed schema file, located in the `\conf` folder.

   Depending on your Solr installation, the location of this folder might be different. The default location is: `C:\solr-7.5\server\solr\sitecore_core_index\conf`

2. In the `<uniqueKey>id</uniqueKey>` node, set the `uniqueKey` value to `_uniqueid` as follows:

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

3. In the `<field>` section, add the field configuration for the `_uniqueid` value as follows:

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

4. Save your changes.

5. In the `\conf\Solrconfig.xml` file, delete the following line:

   ```
   <schemaFactory class="ClassicIndexSchemaFactory" />.
   ```

   Important
   Repeat steps 1–5 for each core index.

8.4.3 Populate the Solr Managed Schema

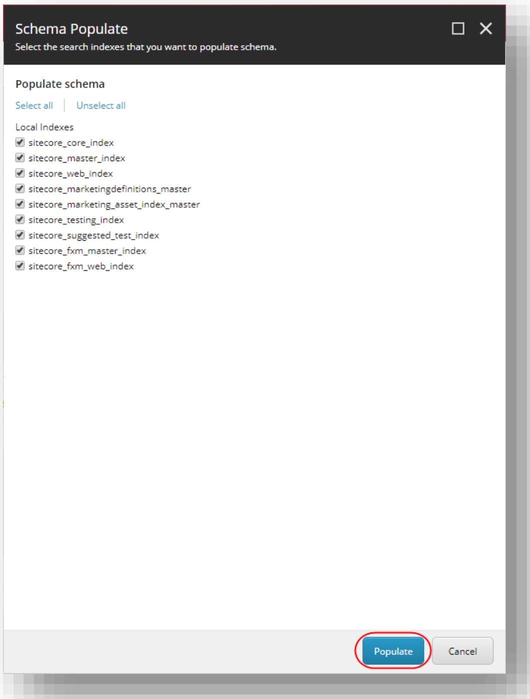
Important
During the installation of the update package, the previous configuration file is disabled. If the previous file had any customizations, a new merged file is created and disabled. If you have any customizations , you should disable the new configuration file, review and enable the newly created 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 the 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

8.5 Update Sitecore.Analytics.ExcludeRobots.config file

In the `\App_Config\Sitecore\Marketing.Tracking\Sitecore.Analytics.ExcludeRobots.config` file, remove the user agents from the `<excludedUserAgents>` node.

Sitecore uses the device detection component to identify robots and the user agents listed here are detected as robots.

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

Chapter 9

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:

9.1 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 Release page. 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.

9.1.1 Cache Configuration

This section is *only* relevant if you have upgraded from any version of Sitecore XP 8.1 to Sitecore XP 9.2.0.

When the upgrade is completed, you must optimize caching to improve performance, reduce server load, and increase system capacity.

For more information, see the topic cache API changes topic on the Sitecore Documentation Site.

For more information about cache configuration, download the Cache Configuration Reference document.

9.1.2 Update personalization rules

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*

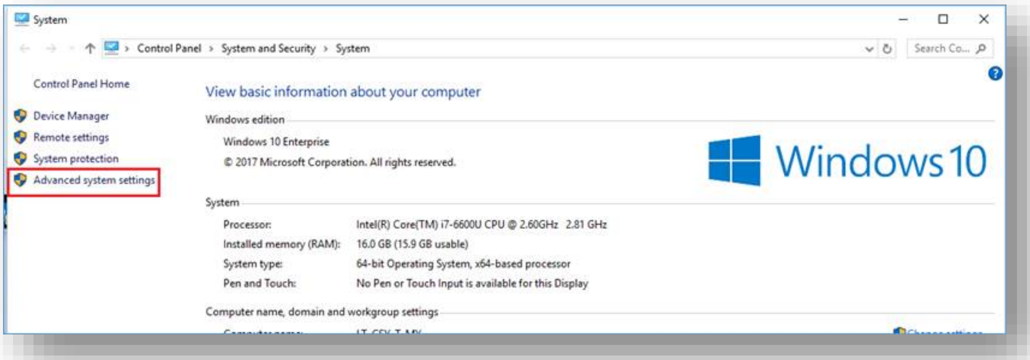
9.2 Enable telemetry reporting in production

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

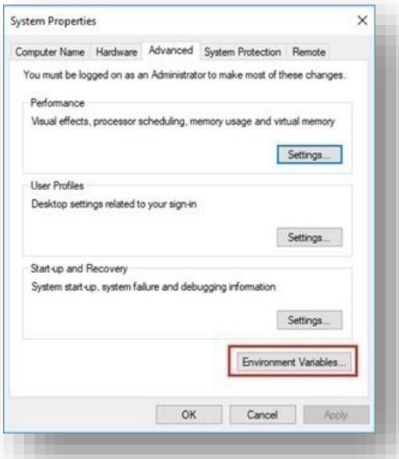
© 2017 Microsoft Corporation. All rights reserved.

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

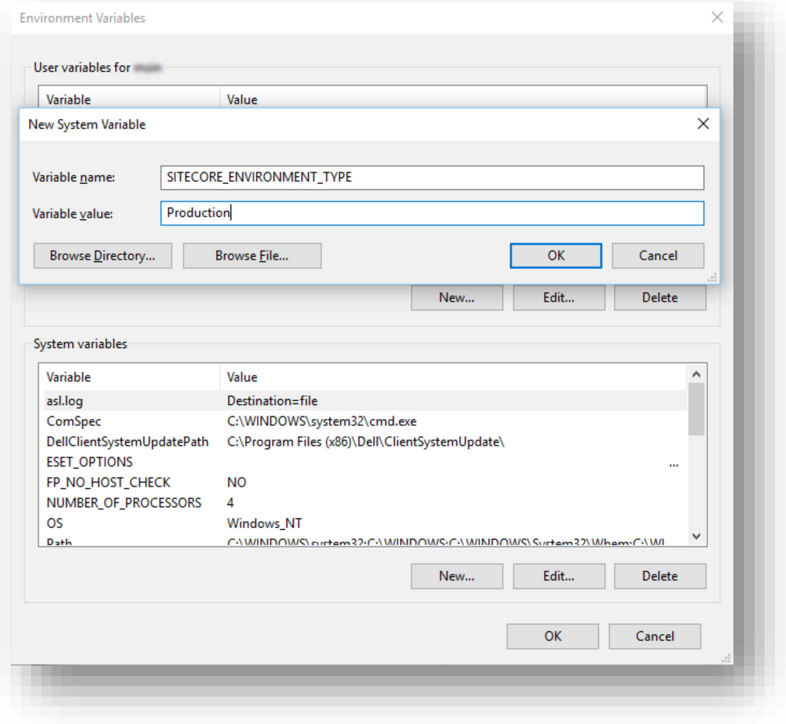
Desktop settings related to your sign-in

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.

9.3 xDB Maintenance

To continue using Sitecore xDB, you must perform all the steps described in this chapter.

9.3.1 Enable xDB and Experience Analytics

You must now enable the Sitecore Experience Database (xDB) and the Experience Analytics functionality that you disabled in the *Disable Experience Analytics* and the *Stop All Content Tests and Disable xDB* sections.

To enable xDB and Experience Analytics:

- In the `\App_Config\Include\Z.Custom` folder, remove the following files:
  - `Disable.xDB.config`
  - `Disable.xAnalytics.config`

9.3.2 Using Email Experience Manager (EXM)

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

If you want to use EXM, you must ensure that it is enabled. It is by default.

In the `\Website\Web.config` file, the `<AppSettings>` section contains the following setting:

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

The following values are supported:

- yes
- no

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.

9.3.3 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.

9.3.4 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, clean the cache using Redis Cache service "Console".
5. Start all the Sitecore XP instances.

9.4 Upgrading from Sitecore XP 8.1 – 8.2.7

If you are upgrading from Sitecore XP 8.1 – 8.2.7, you must perform the tasks described in this section.

9.4.1 Delete Unwanted Marketing Definitions and Marketing Taxonomies

If you are using the xDB functionality, you might have restored some previously deleted marketing definitions and marketing taxonomies before installing the upgrade package.

Now that the upgrade and installation is complete, you can delete these same marketing definitions and marketing taxonomies again.

To see a list of the marketing definitions that were previously deleted:

- In a text editor, in the `\App_Data` folder, open the `RestoredDefinitions_{timestamp}.dat` file.

To see a list of the marketing taxonomies that were previously deleted:

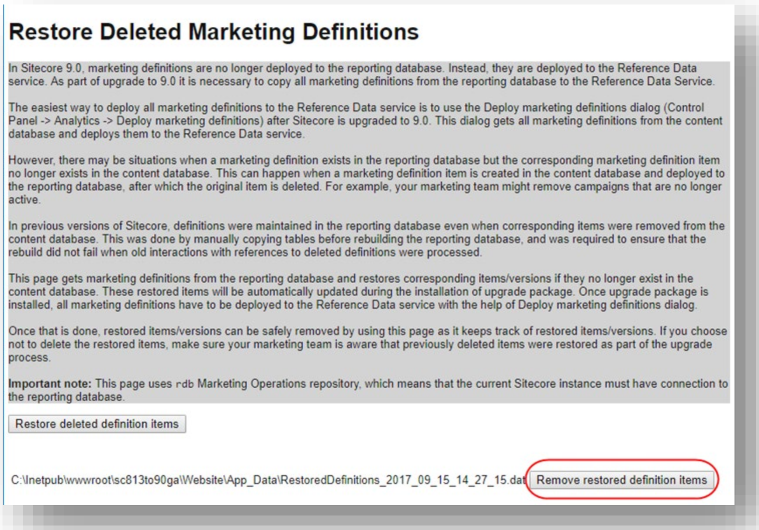
- In a text editor, in the `\App_Data` folder, open the `RestoredTaxonomies_{timestamp}.dat` file.

To delete the unwanted marketing definition items from the *Master* database:

1. To open the **Restore Deleted Marketing Definitions** page, enter the following URL in your web browser:

   http://<hostname>/sitecore/admin/RestoreDeletedMarketingDefinitions.aspx

2. At the bottom of the page, next to the name of the file that contains data about the restored items, click **Remove restored definition items**.



Restore Deleted Marketing Definitions

In Sitecore 9.0, marketing definitions are no longer deployed to the reporting database. Instead, they are deployed to the Reference Data service. As part of upgrade to 9.0 it is necessary to copy all marketing definitions from the reporting database to the Reference Data Service.

The easiest way to deploy all marketing definitions to the Reference Data service is to use the Deploy marketing definitions dialog (Control Panel -> Analytics -> Deploy marketing definitions) after Sitecore is upgraded to 9.0. This dialog gets all marketing definitions from the content database and deploys them to the Reference Data service.

However, there may be situations when a marketing definition exists in the reporting database but the corresponding marketing definition item no longer exists in the content database. This can happen when a marketing definition item is created in the content database and deployed to the reporting database, after which the original item is deleted. For example, your marketing team might remove campaigns that are no longer active.

In previous versions of Sitecore, definitions were maintained in the reporting database even when corresponding items were removed from the content database. This was done by manually copying tables before rebuilding the reporting database, and was required to ensure that the rebuild did not fail when old interactions with references to deleted definitions were processed.

This page gets marketing definitions from the reporting database and restores corresponding items/versions if they no longer exist in the content database. These restored items will be automatically updated during the installation of upgrade package. Once upgrade package is installed, all marketing definitions have to be deployed to the Reference Data service with the help of Deploy marketing definitions dialog.

Once that is done, restored items/versions can be safely removed by using this page as it keeps track of restored items/versions. If you choose not to delete the restored items, make sure your marketing team is aware that previously deleted items were restored as part of the upgrade process.

Important note: This page uses `rdb` Marketing Operations repository, which means that the current Sitecore instance must have connection to the reporting database.

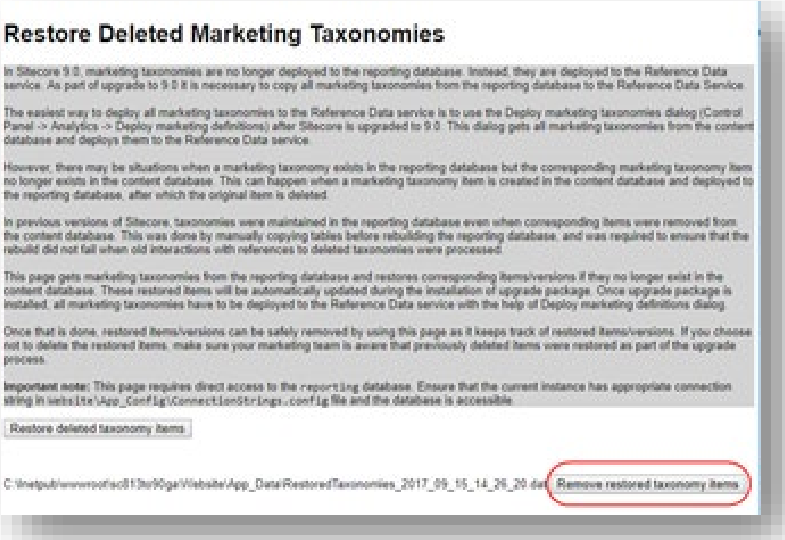
C:\Inetpub\wwwroot\sc813to90ga\Website\App_Data\RestoredDefinitions_2017_09_15_14_27_15.dat

To delete unwanted marketing taxonomy items from the *Master* database:

1. To open the **Restore Deleted Marketing Taxonomies** page, enter the following URL in your web browser:

   http://<hostname>/sitecore/admin/RestoreDeletedMarketingTaxonomies.aspx

2. At the bottom of the page, next to the name of the file that contains data about the restored items, click **Remove restored taxonomy items**.



Restore Deleted Marketing Taxonomies

In Sitecore 9.0, marketing taxonomies are no longer deployed to the reporting database. Instead, they are deployed to the Reference Data service. As part of upgrade to 9.0 it is necessary to copy all marketing taxonomies from the reporting database to the Reference Data Service.

The easiest way to deploy all marketing taxonomies to the Reference Data service is to use the Deploy marketing taxonomies dialog (Control Panel -> Analytics -> Deploy marketing definitions) after Sitecore is upgraded to 9.0. This dialog gets all marketing taxonomies from the content database and deploys them to the Reference Data service.

However, there may be situations when a marketing taxonomy exists in the reporting database but the corresponding marketing taxonomy item no longer exists in the content database. This can happen when a marketing taxonomy item is created in the content database and deployed to the reporting database, after which the original item is deleted.

In previous versions of Sitecore, taxonomies were maintained in the reporting database even when corresponding items were removed from the content database. This was done by manually copying tables before rebuilding the reporting database, and was required to ensure that the rebuild did not fail when old interactions with references to deleted taxonomies were processed.

This page gets marketing taxonomies from the reporting database and restores corresponding items/versions if they no longer exist in the content database. These restored items will be automatically updated during the installation of upgrade package. Once upgrade package is installed, all marketing taxonomies have to be deployed to the Reference Data service with the help of Deploy marketing definitions dialog.

Once that is done, restored items/versions can be safely removed by using this page as it keeps track of restored items/versions. If you choose not to delete the restored items, make sure your marketing team is aware that previously deleted items were restored as part of the upgrade process.

Important note: This page requires direct access to the `reporting` database. Ensure that the current instance has appropriate connection string in `website\App_Config\ConnectionStrings.config` file and the database is accessible.

C:\inetpub\wwwroot\sc813to90ga\Website\App_Data\RestoredTaxonomies_2017_09_15_14_26_20.dat

9.4.2 Update Customizations for Reporting

In Sitecore, the filters in Experience Analytics and the maps in Path Analyzer are based on rule sets that you configure in the Rule Set Editor. In Sitecore XP 9.0, the rule set has been redesigned. If you are upgrading to the latest version of Sitecore from any Sitecore XP 8.x version, you must therefore update all of your custom filters and custom maps by setting and configuring new rules in the Rule Set Editor.

All of the filters and maps that are shipped with Sitecore have been updated already.

If you customized Experience Analytics or created any custom segments, reports, or dashboards, you must update these. If you have created any custom maps in the Path Analyzer, you must also update them.

9.4.3 Upgrade Experience Analytics

In Sitecore XP 9.0.0 or later, some subsystems that affect Experience Analytics have been changed. If you are upgrading from Sitecore XP 8.x and have customized Experience Analytics, you must update these customizations to support the updated subsystems.

Custom Segments

If you have any custom filters or any segments that use these custom filters, you must update these filters manually and redeploy the segments.

The Rule Set Editor has been updated and the rules that it contains have also been changed.

You must therefore:

- Reconfigure any filters that use these custom segments and then redeploy the custom segments.
- Rebuild any custom filters that you have created so that they are compatible with the new subsystems.

For more information about editing filters, see the topic Update a custom filter to use new rules.

Custom Reports

In Sitecore XP 9.0.0 or later, you can collect specific metrics for each dimension. If you created any custom reports or dashboards, you must update these manually, change the dimension/segment that it uses, and select the new version of the metrics that you want to use in the reports or dashboards.

For more information about adding a parameter to a report, see the topic Experience Analytics.

9.4.4 Upgrade Path Analyzer

In the Path Analyzer, you must reconfigure any custom maps that you have created and redeploy all the Path Analyzer maps.

Update Custom Maps

The filter rules have changed. As a result, if you have created any custom maps, you must map these to the new filter rules.

Redeploy the Path Analyzer Maps

In Sitecore XP 9.0 or later, the deployed Path Analyzer maps are stored in a different format. You must therefore redeploy the Path Analyzer maps.

To redeploy the Path Analyzer maps:

1. In SQL Server Management Studio, in the *Reporting* database, to remove all the data from the *TreeDefinitions* table, execute the following command: `Delete from TreeDefinitions`.
2. In Sitecore, open the admin page of the Path Analyzer – `/sitecore/admin/pathanalyzer.aspx`.
3. In the **Maps Manager** section, click **Deploy all maps that are not deployed**.

9.5 Upgrading from Sitecore XP 9.0.1

If you are upgrading from Sitecore XP 9.0.1, you must perform some of the tasks described in this section.

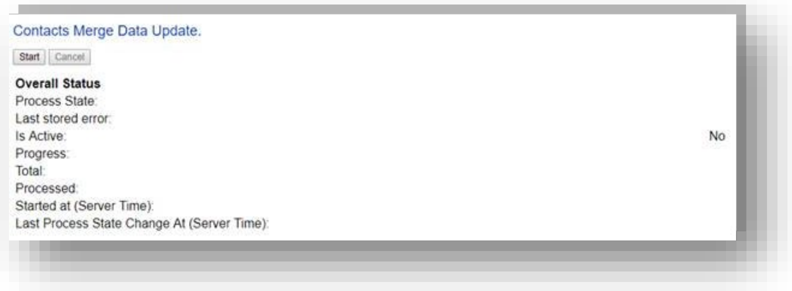
9.5.1 Upgrade the Contact Merge Data

Important
If you are upgrading from Sitecore XP 9.0 or later, you must perform this step.

In Sitecore XP 9.0 Update 2, the contact merge behavior was changed and when you merge contacts, interactions are copied from the source contact to the target contact and the source contact is marked as obsolete.

To apply the new contact merge logic to existing contacts that have previously been merged:

1. On the xDB Processing server, open the `/sitecore/admin/xdb/ContactsMerge.aspx` page.



Contacts Merge Data Update.

Overall Status
Process State:
Last stored error:
Is Active: No
Progress:
Total:
Processed:
Started at (Server Time):
Last Process State Change At (Server Time):

2. Click **Start**.

   This copies all the interactions from the source contact to the target contact and significantly improves the performance of data retrieval.

   When the contact merge upgrade process is complete, the Process State changes to *Completed*.

Note
This process might run for a long time and you can run it in a production environment while the system is live. You can also postpone this step but it is an important part of the upgrade process and you *must* perform it.

If you followed our security recommendation and restricted access to the client from the processing server role, you must enable it to access the *ContactsMerge* page. When the upgrade process is finished, you can restrict access again.

After the merge process completes, summary information is written to the `contactMergeDataUpgrade.log.{date}.txt` file in the `logs` folder. The log file includes status and error information about the merge process. The log is file is always created even in cases where no information is written to it.

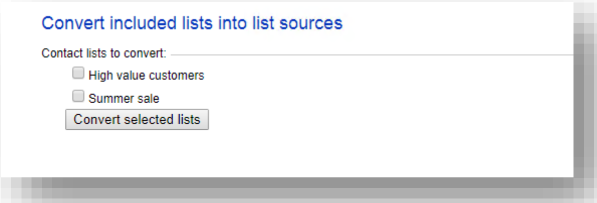
The administration page at `\Website\sitecore\admin\xdb\ContactsMerge.aspx` reports the errors noted in the log file.

9.5.2 Upgrade the Included Lists in List Manager

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

As of 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.

9.5.3 Upgrade the Experience Analytics Data

After you have upgraded Sitecore XP, 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.

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.

Chapter 10

Sitecore Identity Server

Sitecore Identity (SI) is a mechanism that you can use to log 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 uses a separate identity provider, and allows you to set up SSO (Single Sign-On) across Sitecore services and applications.

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

This chapter contains the following sections:

- Install Sitecore Identity Server
- Configuring Sitecore Identity Client

10.1 Install Sitecore Identity Server

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 Sitecore Download page.

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:
   - `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.

10.2 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" />
```

Chapter 11

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.

For a scaled environment, you must repeat the steps in this guide for each instance, with the exception of the actions that are related to the databases, which you must perform only once for each database .

This chapter contains the following section:

11.1 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 the 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 9.0.0 - Sitecore XP 9.0.1 with the Sitecore XP scaled topology, you must perform all the upgrade steps described in the chapter *Upgrade xConnect to Sitecore XP 9.2.0* 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.

To integrate the EXM messaging bus, you must edit the connection strings for the XConnect instance that performs the *Sitecore xConnect MarketingAutomation* role.

Chapter 12

Migrate xDB data

Important
If you are upgrading from Sitecore XP 8.1.0 – 8.2.7 to Sitecore XP 9.2.0, you must perform this step.

If you are upgrading from Sitecore XP 9.0.0 – 9.0.1 to Sitecore XP 9.2.0, skip this step.

In Sitecore XP 9.0.0, we introduced a new data model for the Experience Database (xDB). This means that you must convert all the data in the MongoDB database to the new format. Use the xDB Data Migration Tool to perform this data conversion.

If you want to use the Sitecore Experience Database (xDB), but you do not have to migrate data, then you do not have to use the xDB Data Migration Tool.

You must complete all the procedures in this guide before using the xDB Data Migration Tool.

This chapter contains the following section:

- Use the xDB Data Migration Tool to Migrate xDB data

12.1 Use the xDB Data Migration Tool to Migrate xDB data

Before you run the xDB Data Migration tool, you must deploy the new *Reporting* database to SQL Server.

To deploy the new *Reporting* database:

1. Deploy the `Sitecore.Analytics.dacpac`.

   This file is located in the `Sitecore 9.2.0 rev. 002893 (upgrade files).zip` package, in the `\Databases` folder.

2. On your Sitecore instance, in the `ConnectionStrings.config` file, replace the name of old reporting database with the name of the new one. For example:

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

   You *must* perform this step *once* for every stand alone and Content Management instance.

3. On your xConnect instance, in the `xconnect\App_Config\ConnectionStrings.config` file, replace the name of the old reporting database with the name of the new one. For example:

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

4. On your xConnect instance, in the `xconnect\App_Data\jobs\continuous\ProcessingEngine\App_Config\ConnectionStrings.config` file, replace the name of the old reporting database with the name of the new one. For example:

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

12.1.1 Migrate the xDB Data

After you deploy the new *Reporting* database to SQL Server, you can run the xDB Data Migration tool.

You can download the xDB Data Migration tool from the Sitecore XP 9.0 Downloads page.

For more information about installing and running the xDB Data Migration tool, see xDB Data Migration Tool documentation.

Chapter 13

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.