

Sitecore Experience Platform Upgrade Guide

Sitecore Experience Platform 9.1.1

Upgrade to Sitecore XP 9.1.1 from Sitecore XP 8.1 or later



sitecore[®]
Own the experience[™]

Table of Contents

Chapter 1	Introduction	4
Chapter 2	Prerequisites	5
2.1	Prerequisites	6
2.1.1	Microsoft Visual C++ 2015 Redistributable Requirements	6
2.1.2	Sitecore Experience Commerce	6
2.2	Files and Scripts for Upgrading from Sitecore XP 8.1 or Later	7
Chapter 3	Sitecore Identity Server	9
3.1	Install Sitecore Identity Server	10
Chapter 4	Prepare for the Upgrade	12
4.1	Disable the Sitecore Modules	13
4.1.1	Web Forms for Marketers	13
4.1.2	Email Experience Manager	13
4.1.3	Sitecore Experience Accelerator	13
4.1.4	Data Migration Tool	13
4.2	To Upgrade from Sitecore XP 8.1 – Sitecore XP 8.2 Update 7	14
4.2.1	Restore Deleted Marketing Taxonomies and Marketing Definitions	14
4.2.2	Validate the Names of the Marketing Definition Items	15
4.2.3	Upgrade the Databases	15
4.2.4	Deploy the New Databases	15
4.2.5	Enable Email Experience Manager	16
4.3	To Upgrade from Sitecore XP 9.0 Initial Release or later	18
4.3.1	Disable Experience Analytics	18
4.3.2	Stop the Windows Services	18
4.3.3	Upgrade the Databases	19
4.3.4	Enable Email Experience Manager	19
4.4	Final Preparations	21
4.4.1	Add Security and PackageManagementServiceUrl Connection Strings	21
4.4.2	Indexing and Performance	21
4.4.3	Stop All Content Tests and Disable xDB	21
Chapter 5	Install the Upgrade Package	22
5.1	Install the Upgrade Package	23
5.2	The Upgrade Package Analysis	25
5.2.1	Resolve Breaking Changes in Custom Code	25
5.2.2	Resolve Configuration File Conflicts	26
Chapter 6	Configure Sitecore	30
6.1	Review Changes in Configuration Files	31
6.1.1	The Helix Folder Structure	31
6.1.2	Review patch files with customization	31
6.1.3	Review the Web.config Customizations	31
6.2	Specify the Server Role and the Search Provider	33
6.3	Remove Deprecated Indexes	34
6.4	Update Solr	35
6.4.1	Specify the Solr connecting string	35
6.4.2	Configure Managed Schema	35
6.4.3	Populate the Solr Managed Schema	35
Chapter 7	Install xConnect	37
7.1	Install and Configure Sitecore xConnect	38
Chapter 8	Install the Sitecore Cortex™ Processing Engine	39
8.1	Install and Configure Sitecore Cortex Processing Engine	40
8.2	Install and Configure Microsoft Machine Learning Server	42
Chapter 9	Upgrade xConnect to Sitecore XP 9.1.1	43
9.1	Prepare for the Upgrade	44
9.1.1	Prerequisites for Upgrading xConnect	44

Sitecore Experience Platform Upgrade Guide

9.1.2	Backup your Customizations	44
9.1.3	Stop Windows Services	45
9.2	Upgrade the Databases	46
9.2.1	Deploy the Messaging Database	46
9.2.2	Upgrade existing Databases.....	46
9.2.3	Upgrade the Processing Engine Tasks database.....	47
9.3	Update the Files.....	48
9.3.1	Add New Connection Strings	49
9.3.2	Restore Customizations	49
9.4	Configure the Search Provider	50
9.4.1	Configure Solr	50
9.4.2	Configure Azure Search.....	50
9.5	Configure the Azure Key Vault	52
9.6	Resume the Services.....	53
Chapter 10	Post-upgrade Steps.....	54
10.1	Upgrade Sitecore Identity Server	55
10.2	Configure the Sitecore Identity Client	56
10.3	General Maintenance.....	57
10.3.1	Cache Configuration	57
10.4	Enable Telemetry Reporting	58
10.5	xDB Maintenance.....	60
10.5.1	Enable xDB and Experience Analytics.....	60
10.5.2	Using EXM	60
10.5.3	Redeploy the Marketing Definitions.....	61
10.5.4	Upgrade the Session Database.....	61
10.6	Upgrading from Sitecore XP 8.1 – Sitecore XP 8.2.7	62
10.6.1	Delete Unwanted Marketing Definitions and Marketing Taxonomies.....	62
10.6.2	Update Customizations for Reporting	63
10.6.3	Upgrade Experience Analytics.....	63
10.6.4	Upgrade Path Analyzer	64
10.7	Upgrading from Sitecore XP 9.0, Update 1, or Update 2.....	65
10.7.1	Upgrade the Contact Merge Data.....	65
10.7.2	Upgrade the Included Lists in List Manager	65
10.7.3	Upgrade the Experience Analytics Data	66
Chapter 11	Upgrade Multiple Instances.....	67
11.1	Upgrade a Scaled Environment With Multiple Instances	68
Chapter 12	Migrate xDB data	69
12.1	Use the xDB Data Migration Tool to Migrate xDB data	70
12.1.1	Migrate the xDB Data.....	70
Chapter 13	Breaking Changes from EXM 3.5	71
13.1	Breaking Changes.....	72
13.1.1	Factories Replaced	72
13.1.2	Client API and Messaging Bus	72
Chapter 14	Using the Update Center	73
14.1	Preparing for the Upgrade	74
14.1.1	Steps for Upgrading xDB Functionality.....	74
14.1.2	Disable the Sitecore Modules.....	74
14.1.3	Disable custom dlls	74
14.1.4	Add the Security Connection String	74
14.2	Using the Update Center to Upgrade to Sitecore XP 9.1.1	76
14.2.1	Update the Update Installation Wizard	76
14.2.2	Install the Upgrade Package.....	76
14.3	Troubleshooting	79
Chapter 15	Appendix A: The xConnect Application Preparation Script	80

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2020 Sitecore. All rights reserved.

Chapter 1

Introduction

This guide describes how to upgrade from Sitecore Experience Platform 8.1 or later to Sitecore Experience Platform 9.1.1.

If you are upgrading from Sitecore XP 8.1 rev. 151003 (Initial Release) or later, you can upgrade directly to Sitecore XP 9.1.1.

However, if your solution is based on an earlier version of Sitecore, you must first upgrade to Sitecore XP 8.1 rev. 151003 (Initial Release) before you can upgrade to Sitecore XP 9.1.1.

This guide contains the following chapters:

- **Introduction**
- **Prerequisites**
- **Sitecore Identity Server**
- **Prepare for the Upgrade**
- **Install the Upgrade Package**
- **Configure Sitecore**
- **Install xConnect**
- **Install the Sitecore Cortex™ Processing Engine**
- **Upgrade xConnect to Sitecore XP 9.1.1**
- **Post-upgrade Steps**
- **Upgrade Multiple Instances**
- **Migrate xDB data**
- **Breaking Changes from EXM 3.5**
- **Using the Update Center**
- **Appendix A: The xConnect Application Preparation Script**

Chapter 2

Prerequisites

This chapter describes the prerequisites required to upgrade to Sitecore XP 9.1.1.

This chapter contains the following sections:

- Prerequisites
- Files and Scripts for Upgrading from Sitecore XP 8.1 or Later

2.1 Prerequisites

Before you upgrade to Sitecore XP 9.1.1, 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 9.1.1 Installation Guide](#).

2.1.1 Microsoft Visual C++ 2015 Redistributable Requirements

Sitecore XP 9.0 Update-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 may 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.*

2.1.2 Sitecore Experience Commerce

If you are upgrading to Sitecore Experience Commerce (XC) 9.1, you must prepare some Sitecore XC files before you proceed with the Sitecore Experience Platform (XP) 9.1.1 upgrade.

For more information, see the [Sitecore Experience Commerce 9.1 Upgrade Guide](#).

2.2 Files and Scripts for Upgrading from Sitecore XP 8.1 or Later

To upgrade from Sitecore XP 8.1 or later, you must have:

- An instance of [Sitecore 8.1 rev. 151003 \(Initial Release\) or later](#)
- Sitecore 9.1.1 rev. 002459 (update package).zip

This file contains the following files:

- Sitecore 9.1.1 rev. 002459 (update package for 8.1).zip
- Sitecore 9.1.1 rev. 002459 (update package for 8.2).zip
- Sitecore 9.1.1 rev. 002459 (update package for 9.x).zip

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

- Sitecore 9.1.1 rev. 002459 (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 9.1.zip
- Sitecore Update Installation Wizard 4.0.1 rev. 00153 for 8.x-9.0.zip
- Sitecore 9.1.1 rev. 002459 (upgrade files).zip

This file contains:

- Database Upgrade Script.zip

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

- CMS_core.sql
- CMS_core_master_web.sql
- SXP_exm_master_dispatchqueue.sql
- SXP_processing_tasks.sql
- SXP_reporting.sql
- SXP_reporting_migrate_experience_analytics_data.sql
- SXP_exm_master_save_suppression.sql
- SXP_exm_master_suppressions.sql
- SXP_exm_master_update_suppression.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

Sitecore Experience Platform Upgrade Guide

- o Databases.zip

This file contains:

- o Sitecore.Analytics.dacpac
 - o Sitecore.Exm.master.dacpac
 - o Sitecore.Experienceforms.dacpac
 - o Sitecore.Messaging.dacpac
 - o 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 9.1.1.

Note

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

Chapter 3

Sitecore Identity Server

Important

If you are upgrading from Sitecore XP 9.1 Initial Release, you must upgrade to Sitecore Identity Server 2.0.1 after you have upgraded Sitecore XP. This is described later in this manual.

If you are upgrading from any other version of Sitecore XP, you *must* install the Sitecore Identity Server before you upgrade your solution.

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

This chapter contains the following sections:

- Install Sitecore Identity Server

3.1 Install Sitecore Identity Server

Before you install Sitecore Identity Server, you must install [.NET Core 2.1.3 Runtime](#) or later.

To install Sitecore Identity Server:

1. Create a folder called `c:\resourcefiles`.
2. Download and save the following [WDP packages and SIF resource files](#) in this folder:
 - o `Sitecore.IdentityServer.2.0.1 rev. 00166.scwdp.zip`
 - o `IdentityServer.json`
 - o `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.2.0.1 rev. 00166.scwdp.zip"
    SqlDbPrefix = $prefix
    SqlServer = $SqlServer
    SqlCoreUser = $SqlAdminUser
    SqlCorePassword = $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.

Chapter 4

Prepare for the Upgrade

Important

If you are upgrading from Sitecore XP 9.0.2, you can use new Update Center and skip all the other chapters in this document unless there is a direct reference to them.

For more information about using the Update Center, see the chapter *Using the Update Center*.

Before you upgrade to Sitecore XP 9.1.1, you must do some preparatory tasks.

This chapter describes the preparatory tasks that you must perform for each upgrade path and has 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:

- Disable the Sitecore Modules
- To Upgrade from Sitecore XP 8.1 – Sitecore XP 8.2 Update 7
- To Upgrade from Sitecore XP 9.0 Initial Release or later
- Final Preparations

4.1 Disable the Sitecore Modules

You must disable any Sitecore modules that you are using.

When the upgrade of Sitecore Experience Platform is completed, you must upgrade all the modules.

4.1.1 Web Forms for Marketers

SitecoreXP 9.1.1 does not support the Web Forms for Marketers module. If you are using the Web Forms for Marketers module, you must disable it before you upgrade to Sitecore XP 9.1.1.

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`

4.1.2 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`.

4.1.3 Sitecore Experience Accelerator

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

4.1.4 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 9.1.1. 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.

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

Important

If you are upgrading from Sitecore XP 8.1 – Sitecore XP 8.2 Update 7 to Sitecore XP 9.1.1, you must perform all the steps described in this section.

If you are upgrading from Sitecore XP 9.0 or later to Sitecore XP 9.1.1, you do not have to perform all the steps described in this section.

4.2.1 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:
 - o `RestoreDeletedMarketingTaxonomies.aspx`
 - o `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.

4.2.2 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.

4.2.3 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.

4.2.4 Deploy the New Databases

In Sitecore XP 9.0, the format of all the databases changed and they are now distributed as Data-tier Application Component Packages (DACPAC), and you can find them in the `Sitecore 9.1.1 rev. 002459 (upgrade files).zip` package, 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.

4.2.5 Enable Email Experience Manager

Note

You must manually migrate your EXM data to Sitecore XP 9.x – 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:
 - o `Sitecore.Exm.master.dacpac`
 - o `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:

- o `EXM.InternalApiKey`

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

- o `EXM.CryptographicKey`
- o `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" />
```

4.3 To Upgrade from Sitecore XP 9.0 Initial Release or later

Important

If you are upgrading from Sitecore XP 8.1 – Sitecore 8.2 Update 7 to Sitecore XP 9.1.1, you do not have to perform the steps described in this section.

If you are upgrading from Sitecore XP 9.0 Initial Release or later to Sitecore XP 9.1.1 and are not using the xDB functionality, you do not have to perform the steps described in this section.

If you are upgrading from Sitecore XP 9.0.2, you can use new Update Center and skip all the other chapters in this document unless there is a direct reference to them.

For more information about using the Update Center, see the chapter Using the Update Center to Upgrade to Sitecore XP 9.1.1.

4.3.1 Disable Experience Analytics

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

1. Unpack the `Sitecore 9.1.1 rev. 002459 (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.

4.3.2 Stop the Windows Services

You must stop these Windows services:

- The Sitecore XConnect Search Indexer service.
- The Sitecore Marketing Automation Engine service.
- Sitecore Cortex™ Processing Engine (applicable for Sitecore XP 9.1 Initial Release only)

If you are upgrading from Sitecore XP 9.1 Initial Release, the *Sitecore Cortex Processing Engine* Windows service is already installed and you must stop it before the upgrade.

Before you start the upgrade, we recommend that you:

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

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

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

- Any deferred tasks that are in the *InProgress* state are re-run.

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

4.3.3 Upgrade the Databases

Important

If you are upgrading from Sitecore XP 9.1 Initial Release, skip this section.

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

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

4.3.4 Enable Email Experience Manager

This section contains information about how to upgrade EXM from:

- Sitecore XP 9.0 Initial Release
- Sitecore XP 9.0.1
- Sitecore XP 9.0.2

Upgrade from Sitecore XP 9.0 Initial Release

Note

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

To enable EXM:

1. Deploy the following files to SQL Server:
 - o `Sitecore.Exm.master.dacpac`
 - o `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:

- o `EXM.InternalApiKey`

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

- o `EXM.CryptographicKey`
- o `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 or Sitecore XP 9.0.2

You must execute the following scripts for the *EXM.Master* database:

- `SXP_exm_master_dispatchqueue.sql`
- `SXP_exm_master_suppressions.sql`
- `SXP_exm_master_save_suppression.sql`
- `SXP_exm_master_update_suppression.sql`

Upgrade the *EXM.Master* database from Sitecore XP 9.1 Initial Release

You must execute the following scripts for the *EXM.Master* database:

- `SXP_exm_master_suppressions.sql`
- `SXP_exm_master_save_suppression.sql`
- `SXP_exm_master_update_suppression.sql`

4.4 Final Preparations

Note

If you are upgrading from Sitecore XP 8.1 – Sitecore XP 9.0.2, you must perform all the steps described in this section.

4.4.1 Add Security and PackageManagementServiceUrl Connection Strings

Before you upgrade to Sitecore XP 9.1.1, 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 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;DataSource=(server);Database=Sitecore.Core" />
```

3. Add the *PackageManagementServiceUrl* connection string. If you are upgrading from Sitecore XP 9.0.2 or later, you can skip this step because the connection string is there already.

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

4.4.2 Indexing and Performance

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"/>
```

4.4.3 Stop All Content Tests and Disable xDB

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

Stop All Content Tests

Before you upgrade your Sitecore installation, you must stop any content tests that are currently running. This prevents the start date of the tests from being overwritten with the upgrade date.

Disable xDB

To disable xDB:

1. Unpack the `Sitecore 9.1.1 rev. 002459 (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 5

Install the Upgrade Package

Important

If you are upgrading from Sitecore XP 9.0.2, you can use new Update Center and skip all the other chapters in this document unless there is a direct reference to them.

For more information about using the Update Center, see the chapter *Using the Update Center*.

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

5.1 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**.

Alternatively, you can open the Installation Wizard from the Sitecore Desktop. Enter the following URL in your web browser:

```
http://<hostname>/sitecore/shell/
```

and then click **Sitecore, Development Tools, Installation Wizard**.

2. If you are upgrading from Sitecore XP 8.1 – Sitecore XP 9.0.2, install the Sitecore Update Installation Wizard 4.0.1 rev. 00153 for 8.x-9.0.zip package.

If you are upgrading from Sitecore XP 9.1 Initial Release, install the Sitecore Update Installation Wizard 4.0.1 rev. 00153 for 9.1.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:

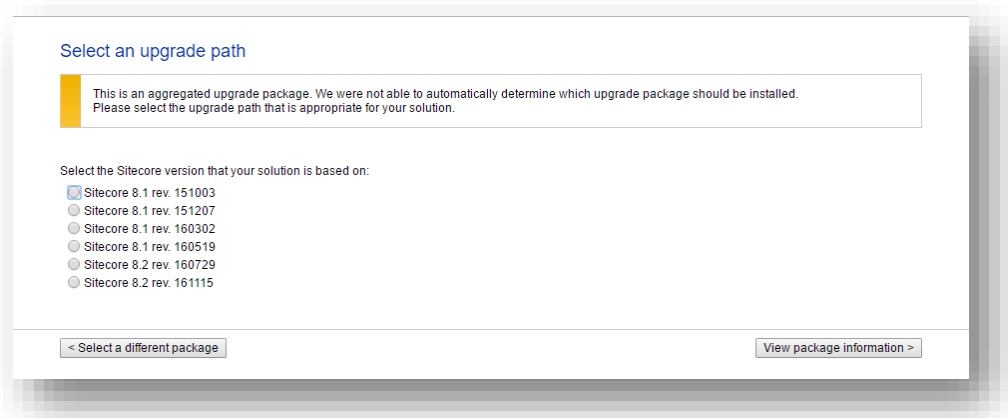
```
http://<hostname>/sitecore/admin/UpdateInstallationWizard.aspx
```

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 9.1.1. Select the update file for the Sitecore version that you are upgrading from, and then click **Open**.
6. Click **Package information**.

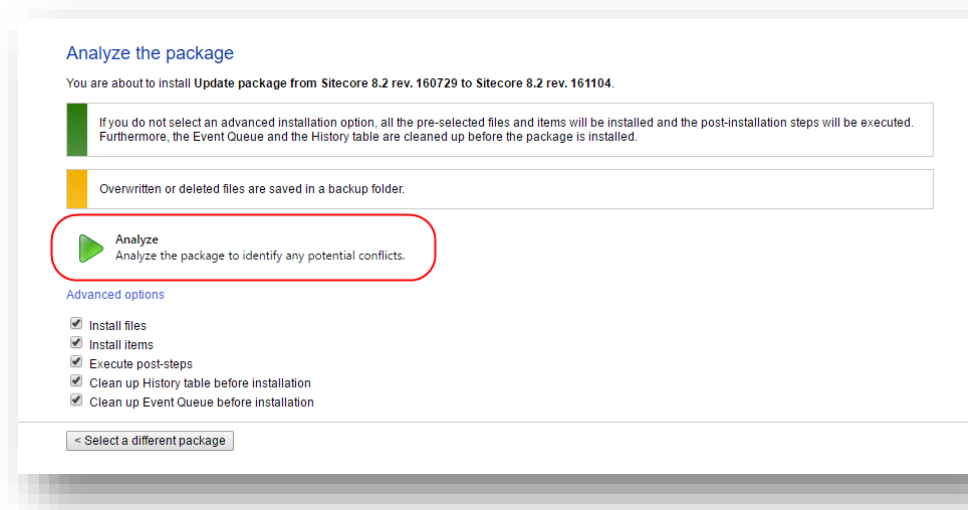
If the **Update Installation Wizard** cannot detect your current Sitecore version automatically, on the **Select an upgrade path** page, you must manually select the Sitecore version and revision number that your solution is based on, and then click **View package information**.



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



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



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*.

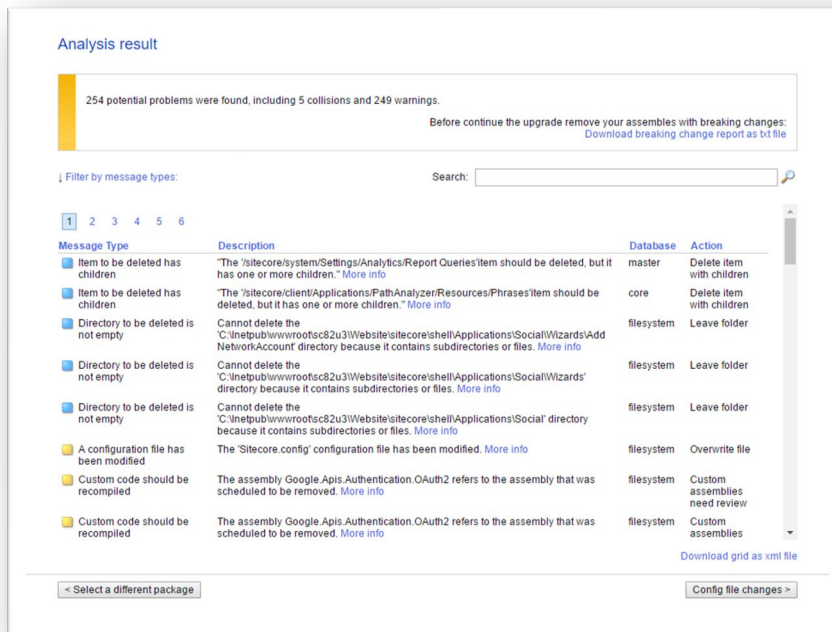
5.2 The Upgrade Package Analysis

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.



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

Important

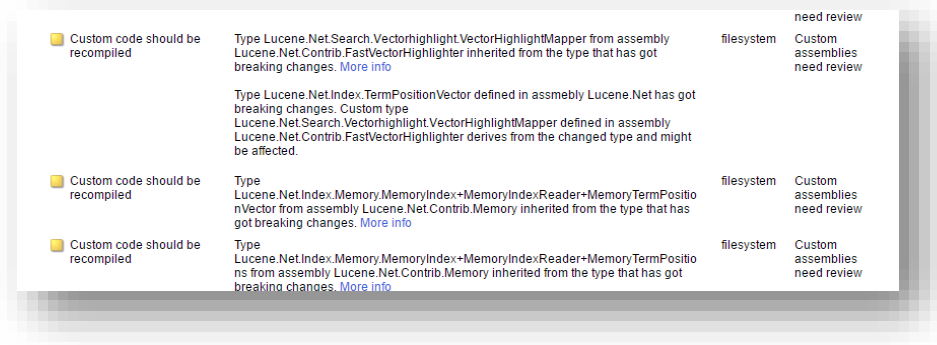
While the majority of the actions ensure the best upgrade experience, if the analysis identifies custom code that must be recompiled or configuration file conflicts that must be resolved, you must address these manually.

5.2.1 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.

Sitecore Experience Platform Upgrade Guide

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



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

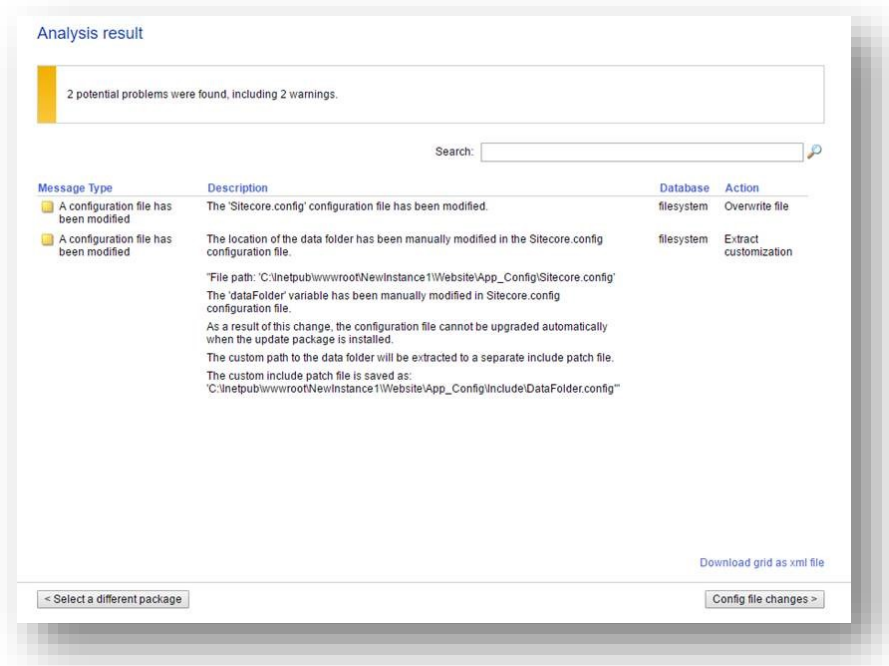
- If you are sure that the affected code is not used during Sitecore startup or in important Sitecore operations such as item management, security management, file management, or in regular website requests that require Sitecore functionality, you can ignore the warnings.

Note

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

5.2.2 Resolve Configuration File Conflicts

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



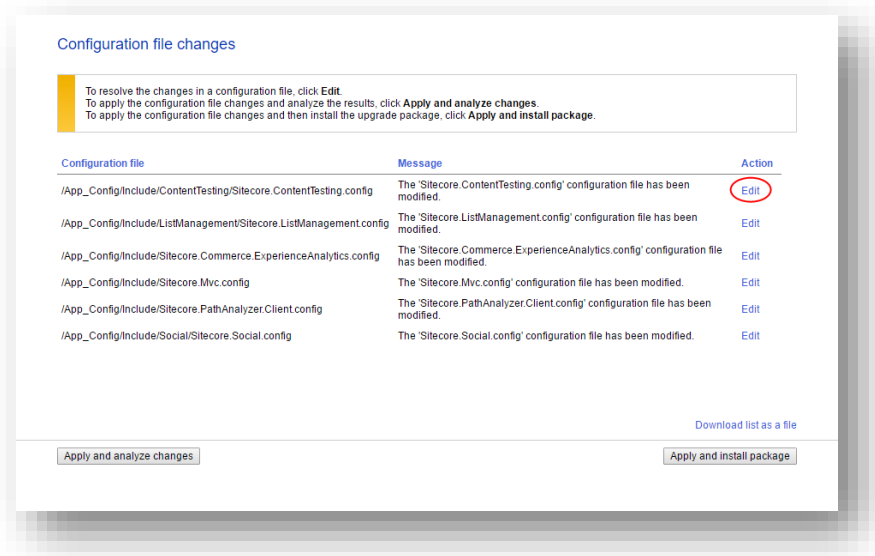
Sitecore Experience Platform Upgrade Guide

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

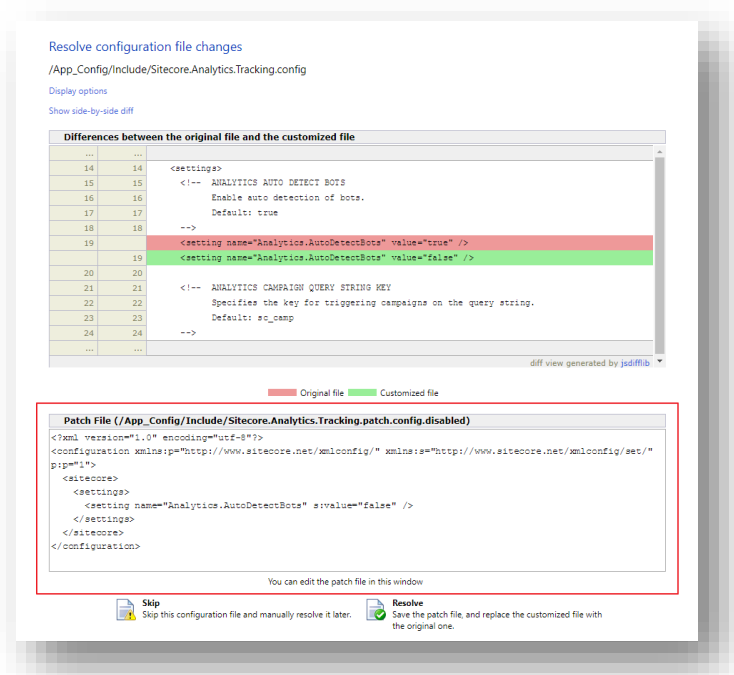
To see more information about a configuration file change, click the description.

To resolve configuration file conflicts:

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



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.



Sitecore Experience Platform Upgrade Guide

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**.

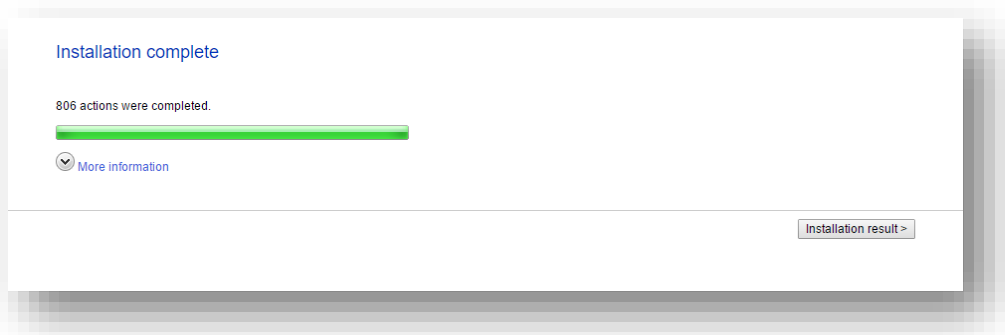
Note

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.

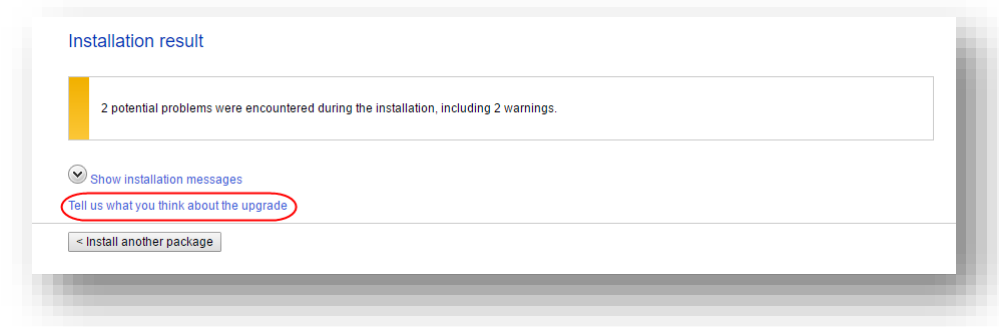
10. When the update process is finished, on the **Installation complete** page, click **Installation result**.



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

Sitecore Experience Platform Upgrade Guide

there was no unexpected data loss. Usually, the problems are inconsequential and the upgrade is successful.



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 6

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:

- Review Changes in Configuration Files
- Specify the Server Role and the Search Provider
- Remove Deprecated Indexes
- Update Solr

6.1 Review Changes in Configuration Files

6.1.1 The Helix Folder Structure

Sitecore XP 9.1.1 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 need certain files to load first, you must place them in the `Website\App_Config\Include\Foundation` folder.

6.1.2 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.

6.1.3 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.

Sitecore Experience Platform Upgrade Guide

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

Since Sitecore XP 9.0, the `web.config` file contains the `role:define` and `search:define` settings that define the Sitecore server role and the search provider. After you upgrade to Sitecore XP 9.1.1, the `web.config` file contains the default values for these settings.

If there were any customizations in this file, they are stored in the `Web.merged.config.disabled` file.

6.2 Specify the Server Role and the Search Provider

Note

If you are upgrading Sitecore XP 9.0 or later, you do not need to specify the server role or the search provider and can skip this section.

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

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
- Processing
- Reporting
- Standalone

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

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
- Solr
- Azure

The default value is *Lucene*.

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

For more information about rule-based configuration, see the [Sitecore Documentation](#).

6.3 Remove Deprecated Indexes

Important

If you are upgrading from Sitecore XP 8.1 – Sitecore 8.2 Update 7 to Sitecore XP 9.1.1, you must perform this step.

If you are upgrading from Sitecore XP 9.0 or later to Sitecore XP 9.1.1, 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\\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**.

6.4 Update Solr

To continue using the Solr search provider in Sitecore XP 9.1.1, you must update your existing Solr instance to version 7.2.1. For more information on how to update Solr, see the [Solr documentation](#).

6.4.1 Specify the Solr connecting string

In Sitecore XP 9.0 Update 2, the Solr connection string was moved to the `ConnectionStrings.config` file.

To continue using Solr after you upgrade to Sitecore XP 9.1.1, you must:

1. Locate the connection string for Solr.
2. Add following connection string to `\Website\App_Config\ConnectionStrings.config` file:

```
<add name="solr.search" connectionString="https://<solr_url>:<solr_port>/solr" />
```

3. Remove the connection string from the old location.

6.4.2 Configure Managed Schema

Since Sitecore XP 9.0, Sitecore supports the Managed Schema approach. Therefore, it is no longer necessary to set up Solr Server to run the Classic Schema.

Important

If you are upgrading from Sitecore XP 9.0 or later, 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. For example, the default location is: `C:\solr-7.2.1\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.

6.4.3 Populate the Solr Managed Schema

Important

During the installation of the update package, all the custom Solr configurations are extracted into patch files. After you have installed the update package, you must review any Solr patch files that were generated and enable all the files that are applicable for your Solr configuration.

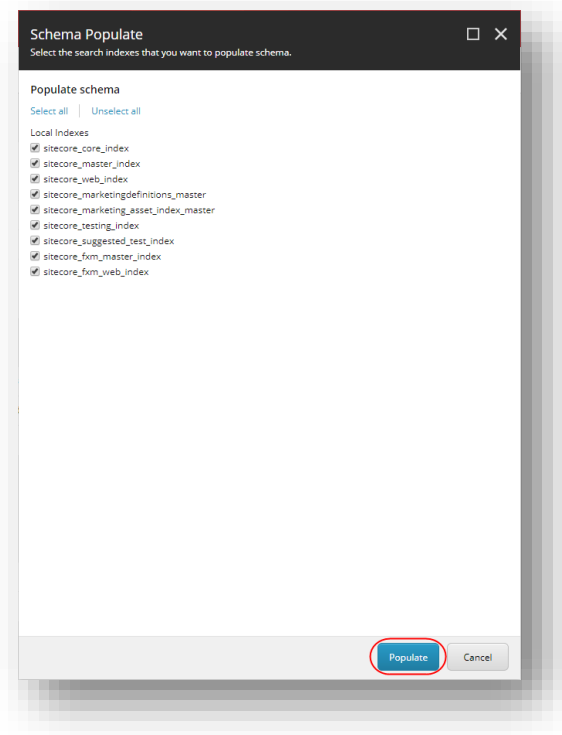
For more information, see the section *Review Changes in Configuration Files*.

Sitecore Experience Platform Upgrade Guide

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

To populate the Solr managed schema:

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



Chapter 7

Install xConnect

If you want to use the Sitecore Experience Database (xDB) functionality in Sitecore XP 9.1.1, you must install and configure Sitecore xConnect.

Important

If you are upgrading from Sitecore XP 8.1 – Sitecore XP 8.2 Update 7, you must install and configure Sitecore xConnect.

If you are upgrading from Sitecore XP 9.0 or later *with* xConnect installed, skip this chapter.

If you are upgrading from Sitecore XP 9.0 or later *without* xConnect installed and want to use the Sitecore Experience Database (xDB) functionality in Sitecore XP 9.1.1, you must perform the tasks described in this chapter.

This chapter contains the following section:

- Install and Configure Sitecore xConnect

7.1 Install and Configure Sitecore xConnect

For more detailed information about which topologies support xConnect, see the *Sitecore Experience Platform 9.1.1 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;FindValue="/>
<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;FindValue=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 8

Install the Sitecore Cortex™ Processing Engine

The Sitecore Cortex Processing Engine is a new role in Sitecore XP 9.1 Initial Release that you can use with or without Microsoft Machine Learning Server (MLS).

Important

If you are upgrading from Sitecore XP 9.1 Initial Release with xConnect installed, the Sitecore Cortex Processing Engine is already installed and you can skip this chapter.

If you are upgrading from Sitecore XP 9.0 – Sitecore XP 9.0.2, you must install and configure Sitecore Cortex Processing Engine.

If you are upgrading from Sitecore XP 9.0 or later without xConnect installed and want to use the Sitecore Experience Database (xDB) functionality in Sitecore XP 9.1.1, you can skip this chapter because you already installed the Sitecore Cortex Processing Engine in chapter 7.

For more detailed information about installing the Sitecore Cortex Processing Engine — including the deployment topologies — see the [Sitecore Experience Platform 9.1.1 Installation Guide](#).

This chapter contains the following sections:

- Install and Configure Sitecore Cortex Processing Engine
- Install and Configure Microsoft Machine Learning Server

8.1 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:

1. [Download the packages for XP Single](#) – Sitecore 9.1.1 rev. 002459 (WDP XP0 packages) .zip – and unpack the zip file.

To install Sitecore Cortex Processing Engine, you need the following file:

- o Sitecore 9.1.1 rev. 002459 (OnPrem)_xp0xconnect.scwdp.zip
2. Unpack the Sitecore 9.1.1 rev. 002459 (OnPrem)_xp0xconnect.scwdp.zip file.
 3. Copy the Sitecore 9.1.1 rev. 002459 (OnPrem)_xp0xconnect.scwdp.zip\Content\Website\App_Data\jobs\continuous\ProcessingEngine folder and all its subfolders and files.
 4. Paste them into the C:\inetpub\wwwroot\instanceName.xconnect\App_Data\jobs\continuous folder.
 5. In the instanceName.xconnect\App_data\jobs\continuous\ProcessingEngine\App_Config\ConnectionStrings.config file, update the following connection strings:
 - o processing.engine.storage
 - o processing.engine.tasks
 - o processing.webapi.blob
 - o processing.webapi.table
 - o xconnect.collection
 - o xconnect.collection.certificate
 - o xconnect.configuration
 - o xconnect.configuration.certificate
 - o xconnect.search
 - o xconnect.search.certificate
 - o messaging
 - o reporting

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

Note

For more information about the `processing.engine.storage`, `processing.engine.tasks`, and the `messaging` connection strings, see the section Upgrade the Databases.

In a scaled environment the `processing.webapi.blob` and the `processing.webapi.table` connection strings should point to the new *Processing Engine* role.

6. In the `xconnect\App_data\jobs\continuous\ProcessingEngine\App_Data` folder, add a valid `license.xml` file
7. 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.

8. 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"
```

8.2 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 9.1.1 Installation Guide](#).

Chapter 9

Upgrade xConnect to Sitecore XP 9.1.1

If you are using the Sitecore Experience Database (xDB) functionality, you must perform all the tasks described in this chapter.

This chapter describes how to upgrade xConnect from any previous version of xConnect – Sitecore XP 9.0.0 or later.

Sitecore XP 9.1 Initial Release introduced the new Sitecore Cortex™ Processing Engine role. The Sitecore Cortex Processing Engine is upgraded as part of xConnect.

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

This chapter contains the following sections:

- Prepare for the Upgrade
- Upgrade the Databases
- Update the Files
- Configure the Search Provider
- Configure the Azure Key Vault
- Resume the Services

9.1 Prepare for the Upgrade

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 9.1.1](#).

9.1.1 Prerequisites for Upgrading xConnect

Before you upgrade xConnect:

1. [Download the packages for XP Single – Sitecore 9.1.1 rev. 002459 \(WDP XP0 packages\)](#) .zip – and unpack the zip file.

To upgrade xConnect, you need the following file:

- o Sitecore 9.1.1 rev. 002459 (OnPrem)_xp0xconnect.scwdp.zip
2. Unpack the Sitecore 9.1.1 rev. 002459 (OnPrem)_xp0xconnect.scwdp.zip file.
 3. Download the [configuration files for upgrade package](#), unpack it, and then unpack the Database Upgrade Script.zip file.

It contains several database upgrade scripts including:

- o SXP_marketingautomation.sql
 - o SXP_referencedata.sql
 - o SXP_processing_pools.sql
 - o SXP_collection.sql
 - o SXP_collection_AE_Part1.sql
 - o SXP_collection_AE_Part2.sql
 - o SXP_collection_GrantPermissions.sql
 - o SXP_processing_engine_tasks.sql
 - o SXP_mongodb_collection.js
4. If you are using Solr provider, ensure that you have updated Solr to version 7.2.1.
 5. If you have any functionality that accesses the contact/interaction data extraction process from outside of xConnect, you must stop it. You can restart it when the upgrade is completed. An example of this kind of functionality is rebuilding the *Reporting* database.

Important

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

If you are upgrading from Sitecore XP 9.1 Initial Release, all the cursors will still work correctly.

9.1.2 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 customizations to a safe location to preserve the files.

9.1.3 Stop Windows Services

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

Ensure that the following Windows services are stopped:

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

9.2 Upgrade the Databases

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

9.2.1 Deploy the Messaging Database

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

1. Deploy the *Messaging* database.
2. In the following files:
 - o \xconnect\App_Config\ConnectionStrings.config
 - o \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.

9.2.2 Upgrade existing Databases

Important

If you are upgrading from Sitecore XP 9.1 Initial Release, skip this section.

You must upgrade the databases on solutions that are running on Sitecore XP 9.0 Initial Release - Sitecore XP 9.0 Update 2.

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 upgrading from Sitecore XP 9.1 Initial Release, skip this section.

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](#).

9.2.3 Upgrade the Processing Engine Tasks database

Important

If you are upgrading from Sitecore XP 9.1 Initial Release, you must upgrade the *Processing Engine Tasks* database.

To upgrade the the *Processing Engine Tasks* database:

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

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

9.3 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:
 - o \Bin
 - o \App_Data\collectiondeployment
 - o \App_Data\Config
 - o \App_Data\Diagnostics
 - o \App_Data\Models
 - o \App_Data\solrcommands
 - o \App_Data\jobs\continuous\AutomationEngine
 - o \App_Data\jobs\continuous\AutomationEngine\AppData\Config
 - o \App_Data\jobs\continuous\AutomationEngine\AppData\Diagnostics
 - o \App_Data\jobs\continuous\IndexWorker
 - o \App_Data\jobs\continuous\IndexWorker\AppData\Config
 - o \App_Data\jobs\continuous\IndexWorker\AppData\Diagnostics
 - o \App_Data\jobs\continuous\IndexWorker\AppData\Models
 - o App_Data\jobs\continuous\ProcessingEngine
 - o App_Data\jobs\continuous\ProcessingEngine\AppData\Config
 - o App_Data\jobs\continuous\ProcessingEngine\AppData\Diagnostics
 - o App_Data\jobs\continuous\ProcessingEngine\AppData\Models
2. In the \App_Data\jobs\continuous\IndexWorker folder, delete the following language folders if they exist:
 - o \de-DE
 - o \es-ES
 - o \fr-FR
 - o \it-IT
 - o \ja-JP
 - o \ko-KR
 - o \pt-BR
 - o \ru-RU
 - o \zh-CN
 - o \zh-TW
3. Copy all the files and folders from the Sitecore 9.1.1 rev. 002459 (OnPrem)_xp0xconnect.scwdp\Content\Website folder that you unpacked earlier, with the exception of the following folders:
 - o \App_Config

- o \App_Data\jobs\continuous\AutomationEngine\App_Config
 - o \App_Data\jobs\continuous\IndexWorker\App_Config
 - o \App_Data\jobs\continuous\ProcessingEngine\App_Config
4. When prompted, replace any files that have the same name.

9.3.1 Add New Connection Strings

Important

If you are upgrading from Sitecore XP 9.1 Initial Release, these connection strings are already added and you can skip this section.

1. Add the processing engine storage connection string to the xConnect instance.

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

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

2. 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" />
```

3. 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;FindValue="/>
```

9.3.2 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.

9.4 Configure the Search Provider

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

9.4.1 Configure Solr

Important

If you are upgrading from Sitecore XP 9.1 Initial release, you can skip this section.

If you installed a new instance of Solr instead of updating the existing one to Solr 7.2.1, you must configure the new Solr cores. If you are upgrading from Sitecore XP 9.0 Update 2 or earlier, you must also configure the existing xConnect Solr cores.

To create new Solr cores, in the Solr admin console, add 2 new Solr cores that are based on the default configuration and use the same base name, for example:

- o `xdbcore`
- o `xdbcore_rebuild`

Important

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

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="http://<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
```

In the following connection string files:

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

Update the `solrCore` connection string with the name of the core that is attached to Solr 7.2.1:

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

9.4.2 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:

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

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

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

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

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

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

In Sitecore XP 9.0.2 and Sitecore XP 9.1.1, 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\continuous\IndexWorker\App_data\config\sitecore` folder, disable the following configuration files by adding the `.disabled` extension:

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

9.5 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](#).

9.6 Resume the Services

1. Recycle the IIS xConnect site Application Pool.
2. Start the *Sitecore xConnect Search Indexer* Windows service.
3. Start the *Sitecore Marketing Automation Engine* Windows service.
4. Start the *Sitecore Cortex Processing Engine* Windows service.
5. Rebuild the xConnect indexes.

If you are upgrading from Sitecore XP 9.1, you do not have to rebuild the xConnect indexes.

Chapter 10

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:

- Upgrade Sitecore Identity Server
- Configure the Sitecore Identity Client
- General Maintenance
- Enable Telemetry Reporting
- xDB Maintenance
- Upgrading from Sitecore XP 8.1 – Sitecore XP 8.2.7
- Upgrading from Sitecore XP 9.0, Update 1, or Update 2

10.1 Upgrade Sitecore Identity Server

Important

If you are upgrading from Sitecore XP 9.1 Initial Release, you must upgrade to Sitecore Identity Server 2.0.1.

If you are upgrading from any other version of Sitecore XP, skip this section. You do not have to update Sitecore Identity Server.

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

To upgrade the Sitecore Identity Server:

1. On the Sitecore Identity Server instance, stop the IIS application pool.
2. Extract the content of the `Sitecore Sitecore Identity Server 2.0.1.zip` file to the existing Sitecore Identity Server directory and replace all the files except:
 - o `Config\production\Sitecore.IdentityServer.Host.xml`
 - o `sitecoreruntime\license.xml`
3. Delete the following files from the root folder:
 - o `Sitecore.IdentityServer.Host.pdb`
 - o `Sitecore.Plugin.IdentityProvider.AzureAd.pdb`
 - o `Sitecore.Plugin.IdentityServer.pdb`
 - o `Sitecore.Plugin.IdentityProviders.pdb`
 - o `Sitecore.Configuration.Factory.pdb`
4. Start the IIS application pool again.

10.2 Configure the Sitecore Identity Client

Important

If you are upgrading from Sitecore XP 9.1 Initial Release, skip this section.

There is no need to upgrade the Sitecore Identity server.

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

After the upgrade, you must configure several 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" />
```


10.3 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 are using a client language other than English, download and import the latest translations from the *Client Translations* section of the [release page](#).

10.3.1 Cache Configuration

This section is *only* relevant if you have upgraded from any version of Sitecore XP 8.1 to Sitecore XP 9.1.1.

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.

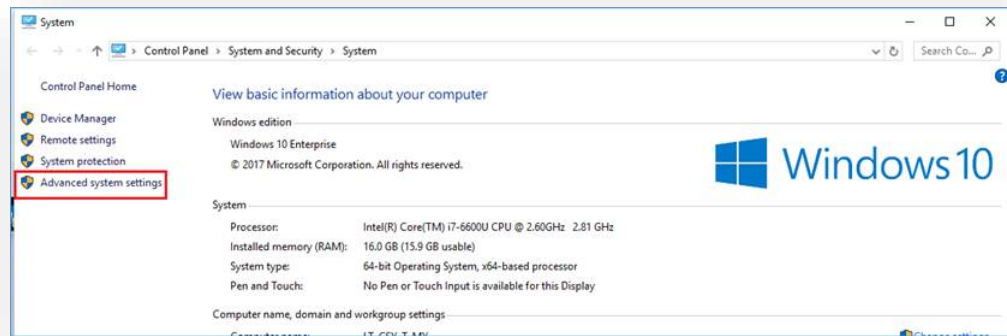
10.4 Enable Telemetry Reporting

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.

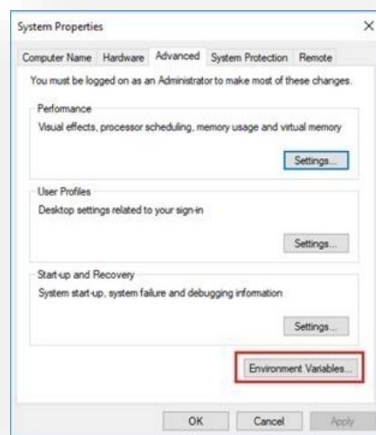
The telemetry system assumes Sitecore is running in a non-production environment by default. When you set up a production environment, you must therefore define a system variable to report the environment type.

To define a system variable:

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



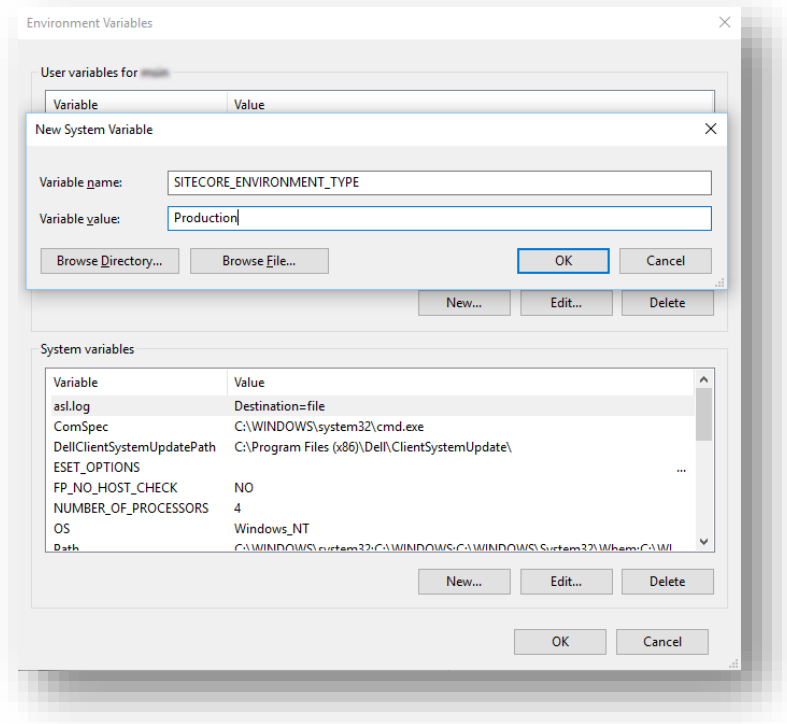
2. In the **System Properties** dialog, click **Environment Variables**.



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

Sitecore Experience Platform Upgrade Guide

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



5. Restart the computer.

10.5 xDB Maintenance

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

10.5.1 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* section.

To enable xDB and Experience Analytics:

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

10.5.2 Using EXM

Important

You can only use EXM when xDB is enabled.

EXM is enabled by default.

To comply with the new rule-based configuration functionality, if you want to use EXM, you must ensure that it is enabled.

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

The default value is yes.

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 configuring email delivery, see [Sitecore Email Cloud compared to the custom SMTP](#).

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](#).

10.5.3 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.

10.5.4 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. Start all the Sitecore XP instances.

10.6 Upgrading from Sitecore XP 8.1 – Sitecore XP 8.2.7

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

10.6.1 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:

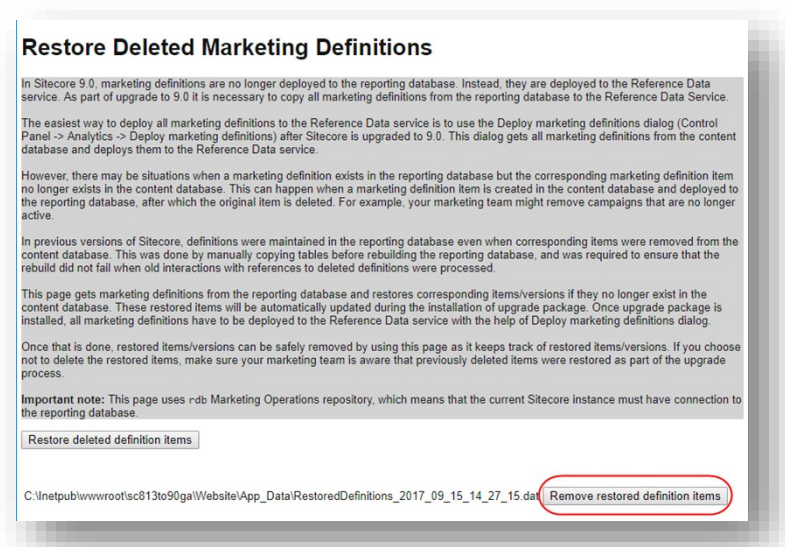
- In a text editor, in the `\App_Data` folder, open the `RestoredTaxonomies_{timestamp}.dat` file.

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**.

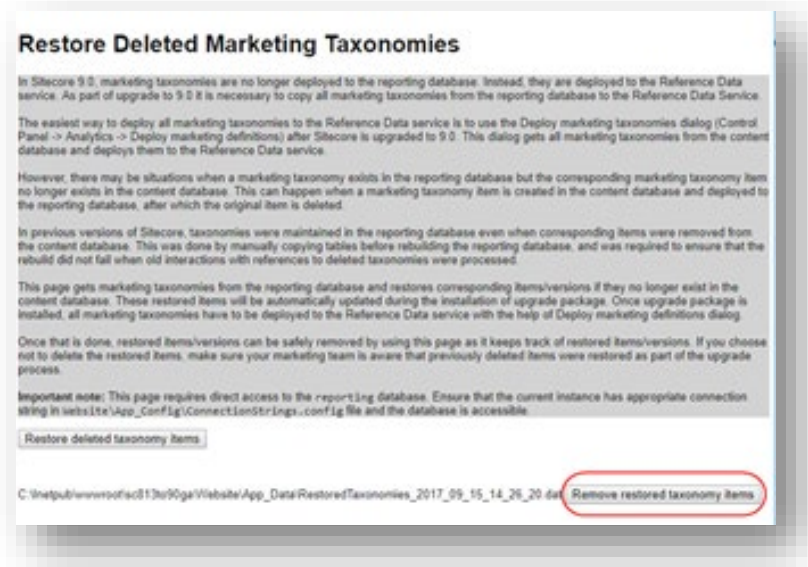


To delete unwanted marketing taxonomy items from the *Master* database:

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**.



10.6.2 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.

10.6.3 Upgrade Experience Analytics

In Sitecore XP 9.0 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 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 [Creating a custom Experience Analytics report](#).

10.6.4 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**.

10.7 Upgrading from Sitecore XP 9.0, Update 1, or Update 2

If you are upgrading from Sitecore XP 9.0 - Update 2, you must perform some of the tasks described in this section.

Important

If you are upgrading from Sitecore XP 9.1 Initial Release, skip this section.

10.7.1 Upgrade the Contact Merge Data

Important

If you are upgrading from Sitecore XP 9.0 Initial Release or Update 1, you must perform this step.

If you are upgrading from Sitecore XP 9.0 Update 2 or later, skip this section.

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.



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 may 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.

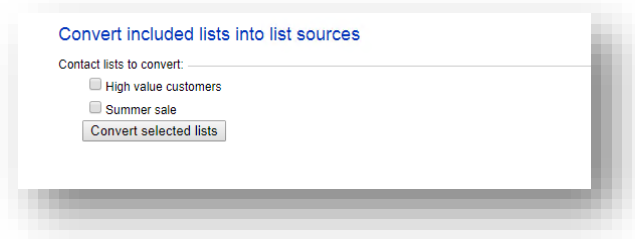
10.7.2 Upgrade the Included Lists in List Manager

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

Sitecore Experience Platform Upgrade Guide

Starting from Sitecore XP 9.1 Initial Release, 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.



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.

10.7.3 Upgrade the Experience Analytics Data

After you upgrade to Sitecore XP 9.1.1, 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:

- o `[dbo].Add_DeviceMetrics_Tvp`
- o `[dbo].Add_GeoMetrics_Tvp`
- o `[dbo].DeviceMetrics_Type`
- o `[dbo].GeoMetrics_Type`
- o `[dbo].Fact_DeviceMetrics`
- o `[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 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 actions related to the database, which you must follow only once for each database .

This chapter contains the following section:

- Upgrade a Scaled Environment With Multiple Instances

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 Content Management servers.

Note

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

Content 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, Content 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 Initial Release - Sitecore XP 9.0 Update 2 with the scaled XP1 XConnect topology, you must perform all the upgrade steps described in the chapter Upgrade xConnect to Sitecore XP 9.1.1 in alignment with the instructions in the [Sitecore scaling scenarios documentation](#).

The Sitecore scaling documentation tells you 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 – Sitecore XP 8.2 Update 7 to Sitecore XP 9.1.1, you must perform this step.

If you are upgrading from Sitecore XP 9.0 or later to Sitecore XP 9.1.1, skip this step.

In Sitecore XP 9.0, we introduced a new data model for the Experience Database (xDB). This means that you must convert all 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 9.1.1 rev. 002459 (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* 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 Downloads site](#).

For more information about installing and running the xDB Data Migration tool, see [xDB Data Migration Tool](#) documentation.

Chapter 13

Breaking Changes from EXM 3.5

This section describes the major breaking changes that you may encounter when you upgrade from EXM 3.5. It is not an exhaustive list of all the changes.

Note

These breaking changes only apply if you are upgrading from EXM 3.5/ Sitecore XP 8.x to Sitecore XP 9.x

This chapter contains the following section:

- Breaking Changes

13.1 Breaking Changes

There are over 1,000 breaking changes, including:

- Approximately 600 changed public methods
- Over 400 removed types
- Approximately 50 public field changes
- Approximately 20 interface changes

13.1.1 Factories Replaced

The following factories have been removed:

- `Sitecore.Modules.EmailCampaign.Factories.BusinessLogicFactory`
- `Sitecore.Modules.EmailCampaign.Factories.EcmFactory`
- `Sitecore.Modules.EmailCampaign.Factories.IoFactory`
- `Sitecore.Modules.EmailCampaign.Factories.GateWayFactory`
- `Sitecore.Modules.EmailCampaign.Factory`

They have been replaced with smaller factories that are responsible for creating only a single type.

You can find most of the new factories in:

- `Sitecore.Modules.EmailCampaign.Factories`
- `Sitecore.EmailCampaign.Cm.Factories`

These factories are registered in the IoC container. You can customize most of the factories by patching the relevant configuration element.

13.1.2 Client API and Messaging Bus

The following files have been removed:

- `Sitecore.Modules.EmailCampaign.Core.ClientApiBase`
- `Sitecore.Modules.EmailCampaign.ClientApi` and `ECMClientService.asmx`

The new messaging bus, shipped with Sitecore XP 9.0.1, eliminates the need for an http endpoint (`ECMClientService.asmx`) on the content management server.

The new `Sitecore.EmailCampaign.Cd.Services.ClientApiService` class replaces the old `ClientApi` class and lets you send automated messages (with custom tokens and/or target languages), subscribing and unsubscribing. When the `ClientApiService` receives a request, a message is submitted to the messaging bus, and then picked up by a registered handler.

Chapter 14

Using the Update Center

If you are running Sitecore XP 9.0.2 or later, you can use the Update Center to upgrade to Sitecore XP 9.1.1.

Before you use the Update Center to upgrade to Sitecore XP 9.1.1, you must have all the prerequisite that are described in this manual in place.

This chapter contains the following sections:

- Preparing for the Upgrade
- Using the Update Center to Upgrade to Sitecore XP 9.1.1
- Troubleshooting

14.1 Preparing for the Upgrade

14.1.1 Steps for Upgrading xDB Functionality

Stop these Windows Services

You must stop these Windows services:

- The Sitecore XConnect Search Indexer service.
- The Sitecore Marketing Automation Engine service.
- Sitecore Cortex™ Processing Engine (applicable for Sitecore XP 9.1 Initial Release only).

For more information about stopping these services, see the section *Stop the Windows Services*.

Stop All Content Tests

Before you upgrade your Sitecore installation, you must stop any content tests that are currently running. This prevents the start date of the tests from being overwritten with the upgrade date.

Upgrade the Databases

To upgrade the databases:

1. Run the `SXP_processing_tasks.sql` script for the *Processing.tasks* database.
2. Run the `SXP_reporting.sql` script for the *Reporting* database.
3. Run the `SXP_exm_master_suppressions.sql` script for the *EXM.Master* database.
4. Run the `SXP_exm_master_save_suppression.sql` script for the *EXM.Master* database.
5. Run the `SXP_exm_master_update_suppression.sql` script for the *EXM.Master* database.
6. If you are for upgrading from Sitecore XP 9.0.2, run the `SXP_exm_master_dispatchqueue.sql` script for the *EXM.Master* database.

14.1.2 Disable the Sitecore Modules

You must disable any Sitecore modules that you use.

For more information about disabling the modules, see the section *Disable the Sitecore Modules*.

14.1.3 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 dlls 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.

14.1.4 Add the Security Connection String

Note

If you are upgrading from Sitecore XP 9.1 Initial Release, skip this section.

If you are upgrading from Sitecore XP 9.0.2, you must perform the all the steps described in this section.

In Sitecore XP 9.1 Initial Release, you can move security from the *Core* database to another database.

Before you upgrade to Sitecore XP 9.1.1, you must add the *security* connection string.

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

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

Sitecore Identity Server

If you are upgrading from Sitecore XP 9.0.2, you must install the Sitecore Identity Server before you upgrade your solution. For more information about installing Sitecore Identity Server, see the chapter *Sitecore Identity Server*.

14.2 Using the Update Center to Upgrade to Sitecore XP 9.1.1

Before you can use the Update Center to install the upgrade package, you must update the Update installation wizard.

14.2.1 Update the Update Installation Wizard

1. On the **Sitecore Launchpad**, click **Control Panel**, and in the **Administration** section, click **Install a package**.

Alternatively, you can open the Installation Wizard from the Sitecore Desktop. Enter the following URL in your web browser:

```
http://<hostname>/sitecore/shell/
```

and then click **Sitecore, Development Tools, Installation Wizard**.

2. If you are upgrading from Sitecore XP 9.0.2, install the Sitecore Update Installation Wizard 4.0.1 rev. 00153 for 8.x-9.0.zip package.

If you are upgrading from Sitecore XP 9.1 Initial Release, install the Sitecore Update Installation Wizard 4.0.1 rev. 00153 for 9.1.zip package.

Note

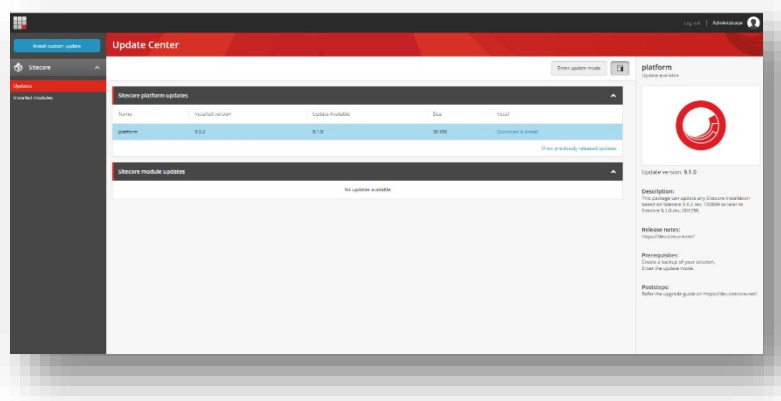
Ensure that Internet Information Services (IIS) is configured to allow access to the `\sitecore\admin` folder.

14.2.2 Install the Upgrade Package

You can now use the Update Center to install the upgrade package.

To install the upgrade package:

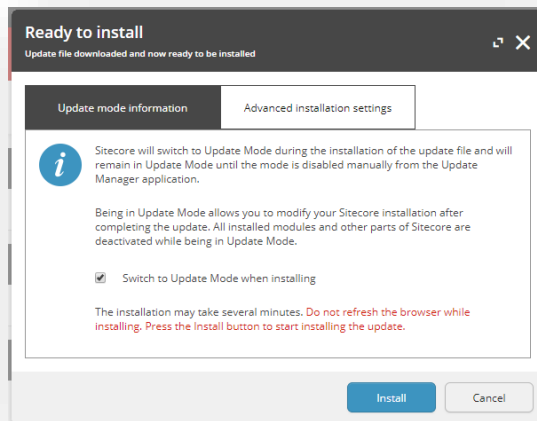
1. Log in to the LaunchPad and open the Update Center.



2. Select the Sitecore XP 9.1.1 upgrade package and then click **Download and Install**

Sitecore Experience Platform Upgrade Guide

3. When the package is downloaded, in the **Ready to install** dialog click **Install**:



4. Wait until the package is installed.

After the upgrade package has been installed, you must configure Sitecore before you exit update mode.

To configure Sitecore, you must:

- Familiarize yourself with new Helix folder structure that is used with Sitecore XP 9.1.1.
- Review all the `.disabled` files that are generated automatically to exclude the customizations that have been made to the Sitecore files, and enable the customizations that you need.
- Review the `Web.config` file.
- Specify the server role and the search provider.
- Update Solr.

For more information about how to configure Sitecore, see the chapter *Configure Sitecore*.

If you used xConnect in Sitecore, you must:

- Upgrade xConnect.

For more information about upgrading xConnect, see the chapters *Install the Sitecore Cortex™ Processing Engine* and *Upgrade xConnect to Sitecore XP 9.1.1*.

If you don't use xConnect and want to configure it after the upgrade, see the chapter *Install xConnect*.

When you completed all these tasks, you must:

- Recompile your solution with the updated Sitecore dlls and fix any breaking changes.
- When you have configured Sitecore, in the Update Center, click **Exit update mode**.



Your Sitecore solution is reloaded and all the configuration changes are enabled.

Sitecore Experience Platform Upgrade Guide

During the upgrade, Sitecore runs in *Update mode* and automatically disables xDB. If you used xDB before the upgrade, it is automatically enabled when you click **Exit update mode**.

- Perform all the post-upgrade steps described in the chapter *Post-upgrade Steps*.

14.3 Troubleshooting

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

I can not see an update to Sitecore XP 9.1.1 in the Update Center

Check the error message that appears in the Update Center.

When I click Install, the installation fails

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

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

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

The update process is divided into three steps:

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

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

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

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

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

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

My Sitecore instance is unstable after being updated

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

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

Ensure that you have setup any additional services that are required for your Sitecore XP 9.1.1 instance to run.

Chapter 15

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.