

# Sitecore Experience Platform Upgrade Guide

Sitecore Experience Platform 9.1 Initial Release

*Upgrade to Sitecore XP 9.1.0 from Sitecore XP 8.1 or later*



**sitecore**<sup>®</sup>  
Own the experience<sup>™</sup>

## 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
2.2	Files and Scripts for Upgrading from Sitecore XP 8.1 or Later .....	7
2.3	Sitecore Identity Server installation .....	9
Chapter 3	Prepare for the Upgrade .....	11
3.1	Disable the Sitecore Modules .....	12
3.1.1	Web Forms for Marketers .....	12
3.1.2	Email Experience Manager .....	12
3.1.3	Sitecore Experience Accelerator .....	12
3.1.4	Data Migration Tool .....	12
3.2	To Upgrade from Sitecore XP 8.1 – Sitecore XP 8.2 Update 7 .....	13
3.2.1	Restore Deleted Marketing Taxonomies and Marketing Definitions .....	13
3.2.2	Validate the Names of the Marketing Definition Items .....	14
3.2.3	Upgrade the Databases .....	14
3.2.4	Deploy the New Databases .....	14
3.2.5	Enable Email Experience Manager .....	15
3.3	To Upgrade from Sitecore XP 9.0 Initial or later .....	17
3.3.1	Disable Experience Analytics .....	17
3.3.2	Stop Two Windows Services .....	17
3.3.3	Upgrade the Databases .....	17
3.3.4	Enable Email Experience Manager .....	17
3.4	Final Preparations for All Versions .....	20
3.4.1	Add Security and PackageManagementServiceUrl Connection Strings .....	20
3.4.2	Indexing and Performance .....	20
3.4.3	Stop All Content Tests and Disable xDB .....	20
Chapter 4	Install the Upgrade Package .....	21
4.1	Install the Upgrade Package .....	22
4.2	The Upgrade Package Analysis .....	24
4.2.1	Resolve Breaking Changes in Custom Code .....	24
4.2.2	Resolve Configuration File Conflicts .....	25
Chapter 5	Configure Sitecore .....	29
5.1	Review Changes in Configuration Files .....	30
5.1.1	The Helix Folder Structure .....	30
5.1.2	Review patch files with customization .....	30
5.1.3	Review the Web.config Customizations .....	30
5.2	Specify the Server Role and the Search Provider .....	32
5.3	Remove Deprecated Indexes .....	33
5.4	Update Solr .....	34
5.4.1	Specify the Solr connecting string .....	34
5.4.2	Configure Managed Schema .....	34
5.4.3	Populate the Solr Managed Schema .....	34
Chapter 6	Install xConnect .....	36
6.1	Install and Configure Sitecore xConnect .....	37
Chapter 7	Upgrade xConnect to Sitecore XP 9.1.0 .....	38
7.1	Prepare for the Upgrade .....	39
7.1.1	Prerequisites for Upgrading xConnect .....	39
7.1.2	Backup your Customizations .....	40
7.1.3	Stop Windows Services .....	40
7.2	Upgrade the Databases .....	41
7.2.1	Deploy Sitecore Cortex™ Processing Engine Databases .....	41
7.2.2	Deploy the Messaging Database .....	41

# Sitecore Experience Platform Upgrade Guide

7.2.3	Upgrade existing Databases.....	42
7.3	Update the Files.....	44
7.3.1	Add New Connection Strings .....	45
7.3.2	Restore Customizations .....	45
7.4	Configure the Search Provider .....	46
7.4.1	Configure Solr .....	46
7.4.2	Configure Azure Search.....	46
7.5	Configure the Azure Key Vault .....	48
7.6	Install the Sitecore Cortex™ Processing Engine .....	49
7.6.1	Install and Configure Microsoft Machine Learning Server .....	49
7.7	Resuming Services.....	52
Chapter 8	Post-upgrade Steps.....	53
8.1	Configure the Sitecore Identity Client .....	54
8.2	General Maintenance.....	55
8.2.1	Cache Configuration .....	55
8.3	xDB Maintenance .....	56
8.3.1	Enable xDB and Experience Analytics.....	56
8.3.2	Using EXM .....	56
8.3.3	Redeploy the Marketing Definitions .....	57
8.3.4	Upgrade the Session Database.....	57
8.4	Upgrading from Sitecore XP 8.1 – Sitecore XP 8.2.7 .....	58
8.4.1	Delete Unwanted Marketing Definitions and Marketing Taxonomies .....	58
8.4.2	Update Customizations for Reporting .....	59
8.4.3	Upgrade Experience Analytics.....	59
8.4.4	Upgrade Path Analyzer .....	60
8.5	Upgrading from Sitecore XP 9.0, Update 1, or Update 2.....	61
8.5.1	Upgrade the Contact Merge Data.....	61
8.5.2	Upgrade the Included Lists in List Manager .....	61
8.5.3	Upgrade the Experience Analytics Data .....	62
Chapter 9	Upgrade Multiple Instances.....	63
9.1	Upgrade a Scaled Environment With Multiple Instances .....	64
Chapter 10	Migrate xDB data .....	65
10.1	Use the xDB Data Migration Tool to Migrate xDB data.....	66
10.1.1	Migrate the xDB Data.....	66
Chapter 11	Breaking Changes from EXM 3.5 .....	67
11.1	Breaking Changes.....	68
11.1.1	Factories Replaced .....	68
11.1.2	Client API and Messaging Bus .....	68
Chapter 12	Using the Update Center .....	69
12.1	Preparing for the Upgrade .....	70
12.1.1	Steps for Upgrading xDB Functionality.....	70
12.1.2	Disable the Sitecore Modules .....	70
12.1.3	Disable custom dlls .....	70
12.1.4	Add the Security Connection String .....	70
12.2	Using the Update Center to Upgrade to Sitecore XP 9.1.0 .....	71
12.2.1	Update the Update Installation Wizard .....	71
12.2.2	Install the Upgrade Package.....	71
12.3	Troubleshooting .....	74
Chapter 13	Appendix A: The xConnect Application Preparation Script .....	75

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 9.1.0.

If you are upgrading from Sitecore XP 8.1 rev. 151003 (Initial Release) or later, you can upgrade directly to Sitecore XP 9.1.0.

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 9.1.0.

This guide contains the following chapters:

- **Introduction**
- **Prerequisites**
- **Prepare for the Upgrade**
- **Install the Upgrade Package**
- **Configure Sitecore**
- **Install xConnect**
- **Upgrade xConnect to Sitecore XP 9.1.0**
- **Post-upgrade Steps**
- **Upgrade Multiple Instances**
- **Migrate xDB data**
- **Breaking Changes from EXM 3.5**
- **Using the Update Center**

## Chapter 2

### Prerequisites

This chapter describes the prerequisites required to upgrade to Sitecore XP 9.1 Initial Release.

This chapter contains the following sections:

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

## 2.1 Prerequisites

Before you upgrade to Sitecore XP9.1.0, you should ensure that you meet the software and hardware requirements for this version of Sitecore XP.

For more information about the requirements, see the Sitecore XP 9.1.0 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.*

## 2.2 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 9.1.0 rev. 001564 (update package).zip

This file contains the following files:

- Sitecore 9.1.0 rev. 001564 (update package for 8.1).zip
- Sitecore 9.1.0 rev. 001564 (update package for 8.2).zip
- Sitecore 9.1.0 rev. 001564 (update package for 9.x).zip

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

- Sitecore 9.1.0 rev. 001564 (update package for 9.x).update

Unpack the appropriate version and save it in a suitable location.

- Sitecore Update Installation Wizard 4.0.0 rev.181002.zip
- Sitecore 9.1.0 rev. 001564 (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

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

- Disable xDB.zip

This file contains:

- Disable.xDB.config
- Disable.xAnalytics.config

- MarketingDefinitionsUpgrade.zip

This file contains:

- RestoreDeletedMarketingTaxonomies.aspx
- RestoreDeletedMarketingDefinitions.aspx
- DefinitionItemsNameValidator.aspx

- Databases.zip

This file contains:

- Sitecore.Analytics.dacpac
- Sitecore.Exm.master.dacpac
- Sitecore.Experienceforms.dacpac

# Sitecore Experience Platform Upgrade Guide

- 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 9.1.0.

## Note

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



## 2.3 Sitecore Identity Server installation

You must install the Sitecore Identity Server before you upgrade your solution. You must use SIF to install Sitecore Identity Server. For more information about how to use SIF, see the [Installation Guide](#).

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

To install Sitecore Identity Server:

1. Create a folder called `c:\resourcefiles`.
2. Download and save the following [WDP packages and SIF resource files](#) in this folder:
  - o `Sitecore.IdentityServer.2.0.0-r00157.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.2.0.0 r00157.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 3

### 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 9.1.0, 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 or later
- Final Preparations for All Versions

## 3.1 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.

### 3.1.1 Web Forms for Marketers

SitecoreXP 9.1.0 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 9.1.0.

To disable the Web Forms for Marketers module:

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

### 3.1.2 Email Experience Manager

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

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

### 3.1.3 Sitecore Experience Accelerator

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

### 3.1.4 Data Migration Tool

If you used the Data Migration Tool 2.0 to migrate your data to Sitecore XP 9.0, you must uninstall the tool before upgrading to Sitecore XP 9.1.0. After you have migrated your data, you no longer need the tool.

For more information about how to uninstall the Data Migration Tool, see the [Data Migration Tool Installation Guide](#).

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

## 3.2 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 9.1.0, you must perform all the steps described in this section.

If you are upgrading from Sitecore XP 9.0 or later to Sitecore XP 9.1.0, you do not have to perform all the steps described in this section.

### 3.2.1 Restore Deleted Marketing Taxonomies and Marketing Definitions

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

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

Before you restore anything:

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

## 3.2.2 Validate the Names of the Marketing Definition Items

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

To validate the names of the marketing definition items:

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

The names of the definition items are validated, and a report called `DefinitionItemsNameValidationErrors_{timestamp}.dat` is saved in the `\App_Data` folder. This report lists all the validation errors that occurred.

5. Ensure you manually fix all the issues that were found, and then validate the names again.  
Repeat this procedure until all the names have been validated.

## 3.2.3 Upgrade the Databases

To upgrade the databases:

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

## 3.2.4 Deploy the New Databases

In Sitecore XP 9.0, the format of all the databases changed and they are now distributed as Data-tier Application Component Packages (DACPAC), and you can find them in the `Sitecore 9.1.0 rev. 001564 (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.

### 3.2.5 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" />
```



## 3.3 To Upgrade from Sitecore XP 9.0 Initial or later

### Important

If you are upgrading from Sitecore XP 8.1 – Sitecore 8.2 Update 7 to Sitecore XP 9.1.0, 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 9.1.0 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*.

### 3.3.1 Disable Experience Analytics

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

1. Unpack the `Sitecore 9.1.0 rev. 001564 (upgrade files).zip` file and the `Disable xDB.zip` file and locate the `Disable.xAnalytics.config` file.
2. In the `\App_Config\Include` folder, create a `\Z.Custom` folder.
3. Copy the `Disable.xAnalytics.config` file to the `\App_Config\Include\Z.Custom` folder.

### Important

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

### 3.3.2 Stop Two Windows Services

You must stop these Windows services:

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

### 3.3.3 Upgrade the Databases

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

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

### 3.3.4 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 Sitecore XP 9.0.2, 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 from Sitecore XP 9.0.1 or Sitecore XP 9.0.2

To upgrade the *EXM.Master* database, you must:

- Execute the `SXP_exm_master_dispatchqueue.sql` script for the *EXM.Master* database.

## 3.4 Final Preparations for All Versions

### 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.

### 3.4.1 Add Security and PackageManagementServiceUrl Connection Strings

Before you upgrade to Sitecore XP 9.1.0 from any version of Sitecore XP, 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:

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

### 3.4.2 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"/>
```

### 3.4.3 Stop All Content Tests and Disable xDB

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

#### Stop 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 9.1.0 rev. 001564 (upgrade files).zip` file and the `Disable xDB.zip` file.
2. Create a `\App_Config\Include\Z.Custom` folder, and copy the `Disable.xDB.config` file to this folder.

#### Important

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

## Chapter 4

### 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 Upgrade to Sitecore XP 9.1*.

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

## 4.1 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. Install the Sitecore Update Installation Wizard 4.0.0 rev.181002.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:

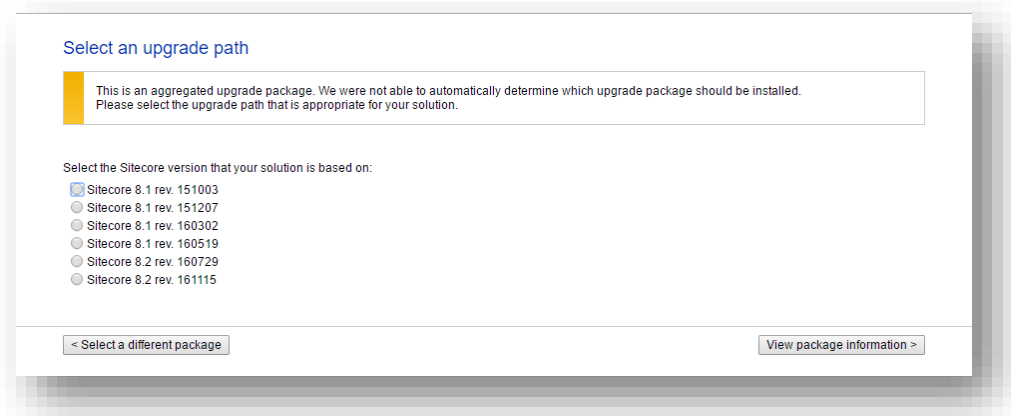
```
http://<hostname>/sitecore/admin/UpdateInstallationWizard.aspx
```

### Note

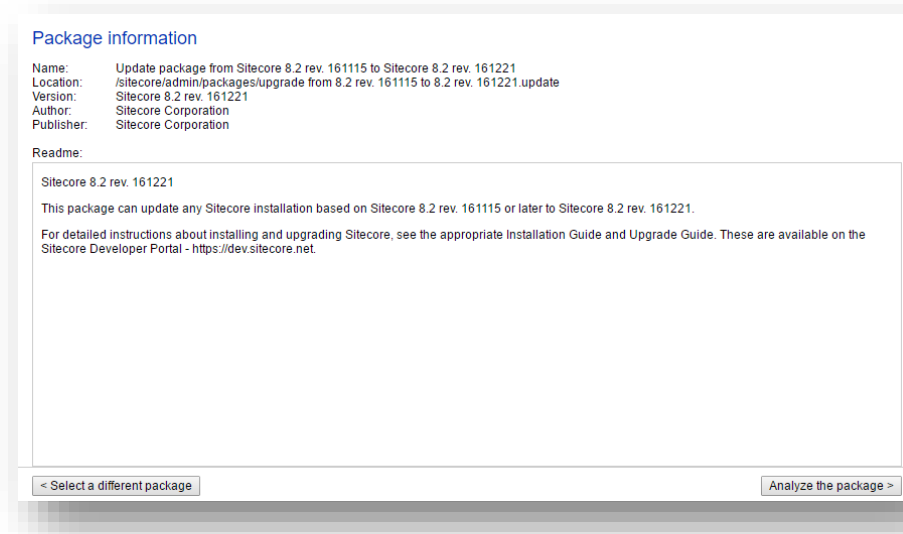
If you are running any version of Sitecore XP 8.1, use the URL to open the wizard.

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

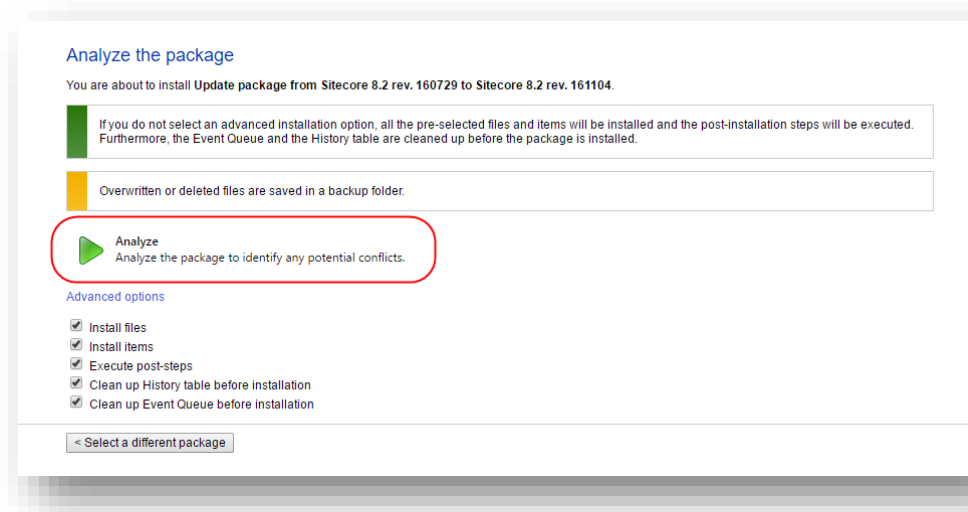
If the **Update Installation Wizard** cannot detect your current Sitecore version automatically, on the **Select an upgrade path** page, you must manually select the Sitecore version and revision number that your solution is based on, and then click **View package information**.



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*.

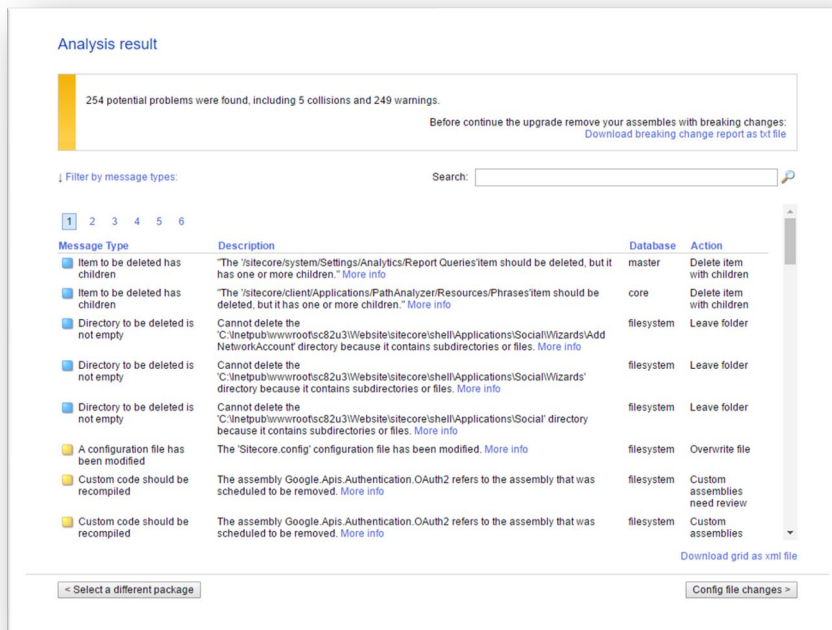
## 4.2 The Upgrade Package Analysis

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.

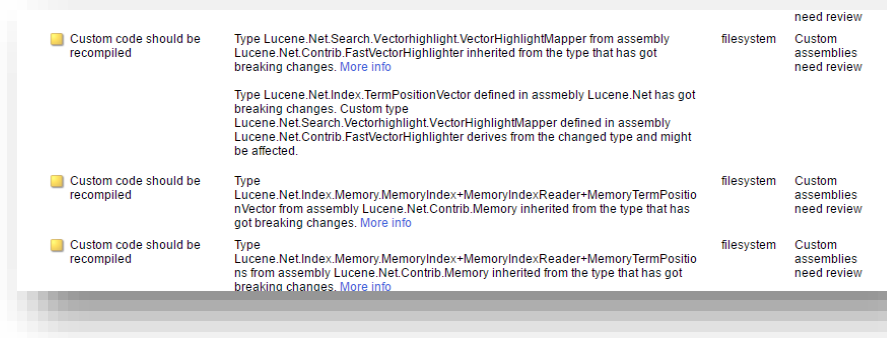
### 4.2.1 Resolve Breaking Changes in Custom Code

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

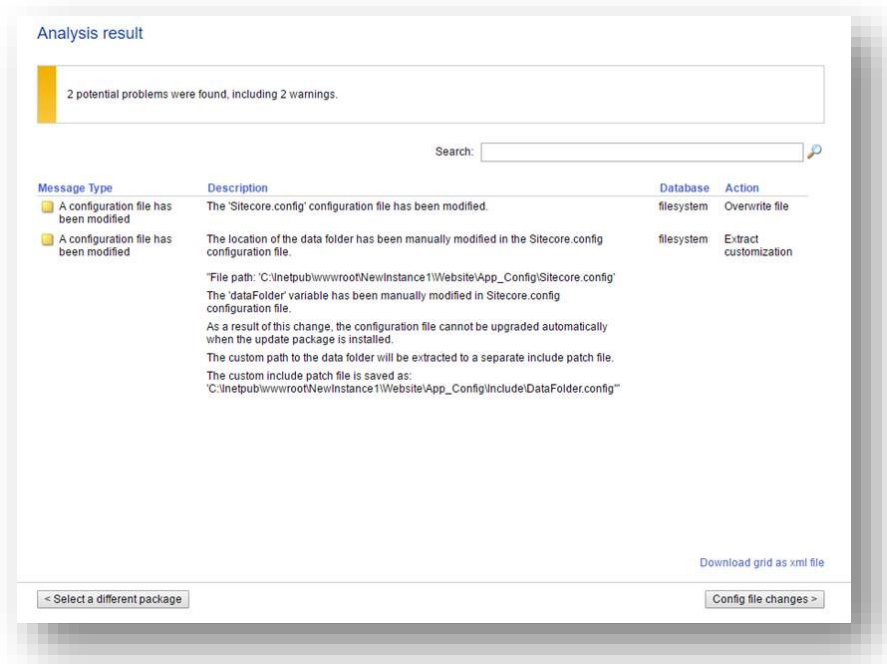
- If you are sure that the affected code is not used during Sitecore startup or in important Sitecore operations such as item management, security management, file management, or in regular website requests that require Sitecore functionality, you can ignore the warnings.

## Note

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

## 4.2.2 Resolve Configuration File Conflicts

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



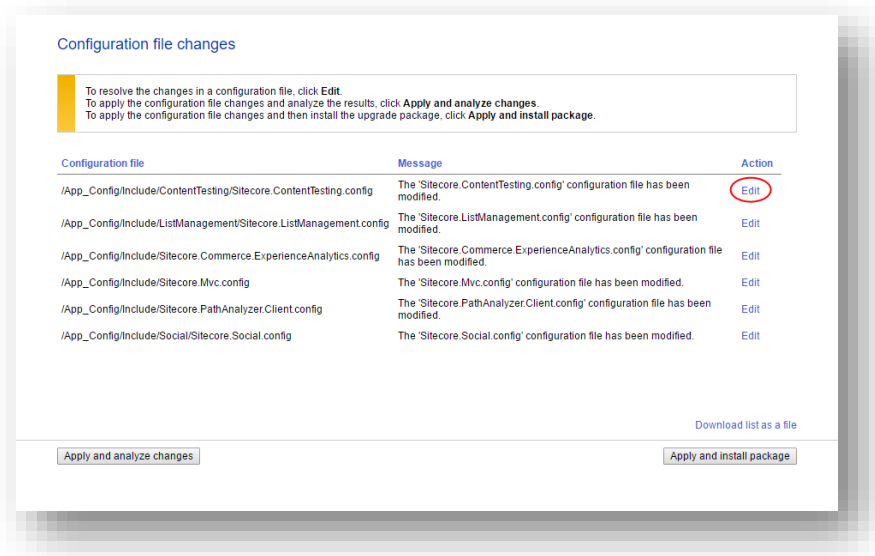
# Sitecore Experience Platform Upgrade Guide

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

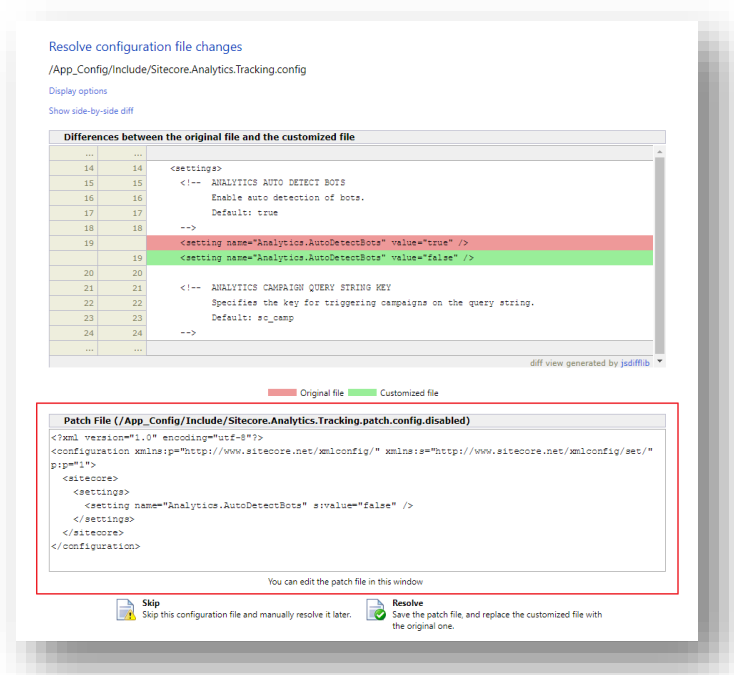
To see more information about a configuration file change, click the description.

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**.

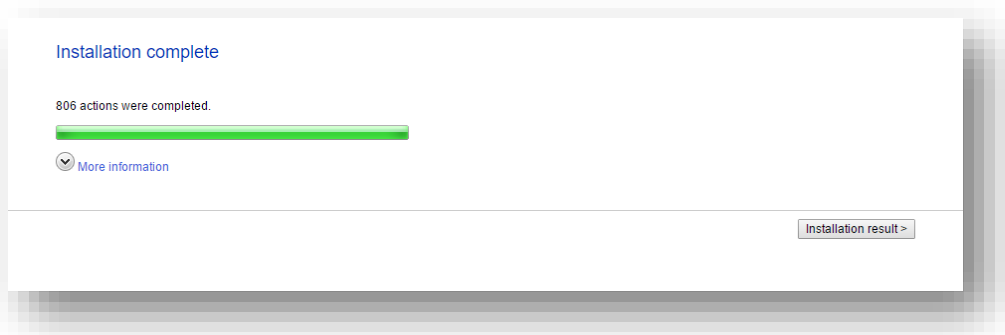
## Note

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.

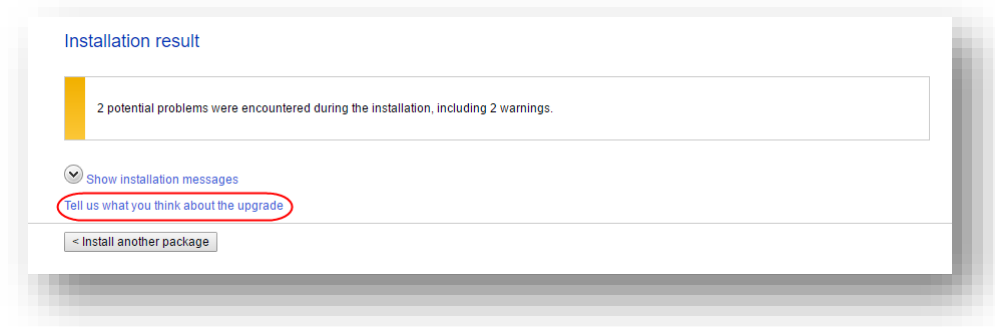
10. When the update process is finished, on the **Installation complete** page, click **Installation result**.



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 5

### 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

## 5.1 Review Changes in Configuration Files

### 5.1.1 The Helix Folder Structure

Sitecore XP 9.1.0 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.

### 5.1.2 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.

### 5.1.3 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 9.1.0, 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.

## 5.2 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](#).



## 5.3 Remove Deprecated Indexes

### Important

If you are upgrading from Sitecore XP 8.1 – Sitecore 8.2 Update 7 to Sitecore XP 9.1.0, you must perform this step.

If you are upgrading from Sitecore XP 9.0 or later to Sitecore XP 9.1.0, skip this step.

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

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

To remove the indexes in Lucene:

1. Navigate to `C:\inetpub\wwwroot\\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**.

## 5.4 Update Solr

To continue using the Solr search provider in Sitecore XP 9.1.0, you must update your existing Solr instance to version 7.2.1. For more information on how to update Solr, see the [Solr documentation](#).

### 5.4.1 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 9.1.0, 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.

### 5.4.2 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.

### 5.4.3 Populate the Solr Managed Schema

#### Important

During the installation of the update package, all the custom Solr configurations are extracted into patch files. You must ensure that all of the patches have been enabled.

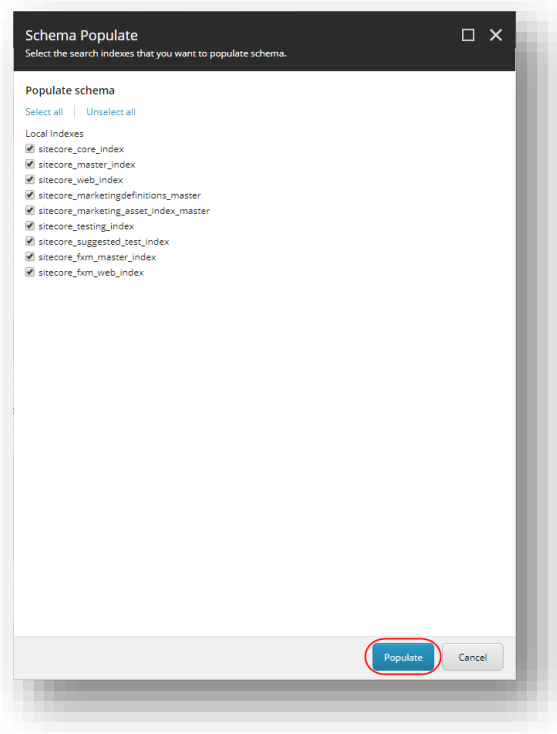
For more information, see the section *Review Changes in Configuration Files*.

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

# Sitecore Experience Platform Upgrade Guide

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 6

### Install xConnect

If you want to use the Sitecore Experience Database (xDB) functionality in Sitecore XP 9.1.0, 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 – Sitecore XP 9.0.2 with xConnect installed, skip this chapter.

If you are upgrading from Sitecore XP 9.0 – Sitecore XP 9.0.2 *without* xConnect installed and want to use the Sitecore Experience Database (xDB) functionality in Sitecore XP 9.1.0, you must perform the tasks described in this chapter.

This chapter contains the following section:

- Install and Configure Sitecore xConnect

## 6.1 Install and Configure Sitecore xConnect

For more detailed information about installing xConnect — including deployment topologies — see the *Sitecore Experience Platform 9.1.0 Installation Guide*. You can download the latest Sitecore Installation Guide from the [Sitecore Downloads Page](#).

To configure Sitecore xConnect for a single instance Sitecore solution:

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

```
<add name="xconnect.collection" connectionString="https://Sitecore_xconnect"/>
<add name="xconnect.collection.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;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 7

### Upgrade xConnect to Sitecore XP 9.1.0

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 Initial release
- Sitecore XP 9.0 Update 1
- Sitecore XP 9.0 Update 2

Sitecore XP 9.1.0 introduces the new Sitecore Cortex™ Processing Engine role. This chapter describes how to install it and connect it to the other roles.

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
- Install the Sitecore Cortex™ Processing Engine
- Resume the Windows Services

## 7.1 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 9.1.0](#).

### 7.1.1 Prerequisites for Upgrading xConnect

Before you upgrade xConnect:

1. [Download the packages for XP Single](#) – Sitecore 9.1.0 rev. 001564 (WDP XP0 packages) .zip – and unpack the zip file.

This package contains three files. To upgrade xConnect, you need the following file:

- o Sitecore 9.1.0 rev. 001564 (OnPrem)\_xp0xconnect.scwdp.zip

2. Unpack the Sitecore 9.1.0 rev. 001564 (OnPrem)\_xp0xconnect.scwdp.zip file.
3. In the Sitecore 9.1.0 rev. 001564 (OnPrem)\_xp0xconnect root folder, there are:

Dacpac files for the 2 new databases:

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

Scripts that create SQL users with the appropriate access rights to the new databases:

- o createuser\_text\_processingenginestorage.sql
- o createuser\_text\_processingenginetaasks.sql

4. 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\_mongodb\_collection.js

5. If you are using Solr provider, ensure that you have updated Solr to version 7.2.1.
6. 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

You cannot just pause these tasks before the upgrade and resume them afterwards because the extraction cursors have changed in Sitecore XP 9.1.0.

## 7.1.2 Backup your Customizations

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

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

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

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

## 7.1.3 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



## 7.2 Upgrade the Databases

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

### 7.2.1 Deploy Sitecore Cortex™ Processing Engine Databases

You must deploy the new databases and add the corresponding connection strings.

1. Deploy the following files to SQL Server:
  - o Sitecore.Processing.engine.storage.dacpac
  - o Sitecore.Processing.engine.tasks.dacpac
2. To create a SQL user with the appropriate access rights for both of the new databases, execute the following scripts:
  - o createuser\_text\_processingenginestorage.sql
  - o createuser\_text\_processingenginetaasks.sql

Execute the appropriate scripts for the *Sitecore.Processing.engine.storage* and *Sitecore.Processing.engine.tasks* databases.

Before you execute the scripts, you must modify each script.

The values that you must modify are highlighted in the following example:

```
EXEC sp_configure 'contained database authentication', 1
GO
RECONFIGURE
GO

ALTER DATABASE [Sitecore.Processing.engine.storage]
SET containment = partial
GO

CREATE USER processingengineuser WITH PASSWORD = 'Password12345';
GO

EXEC sp_addrolemember 'db_datareader', processingengineuser;
EXEC sp_addrolemember 'db_datawriter', processingengineuser;
GO

GRANT EXECUTE TO processingengineuser;
GO
```

3. In the `\xconnect\App_Config\ConnectionStrings.config` file, add the connection string for the *Sitecore.Processing.engine.storage* database:

```
<add name="processing.engine.storage" connectionString="user
id=processingengineuser;password=Password12345;data source=source;
Database=Sitecore.Processing.engine.storage" />
```

#### Note

The connection string for the *Sitecore.Processing.engine.tasks* database is added later when you configure the *Sitecore Cortex™ Processing Engine* windows service.

### 7.2.2 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:XConnect:MarketingAutomation* role and the *Sitecore Cortex™ Processing Engine* role.

## 7.2.3 Upgrade existing Databases

You must upgrade the databases on solutions that are running on Sitecore XP 9.0 Initial Release or later.

### 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

## Note

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 9.1.0 describes how to configure the MongoDB provider for xConnect.

# Sitecore Experience Platform Upgrade Guide

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*.

## 7.3 Update the Files

Now you can update the files on the xConnect instance.

### Note

You can use a Windows batch script to automate steps 1-3. For more information about the batch script, see Appendix A: The xConnect Application Preparation Script.

To update the files:

1. Delete all the files and folders in the following folders.

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\App\_Data\Config
- o \App\_Data\jobs\continuous\AutomationEngine\App\_Data\Diagnostics
- o \App\_Data\jobs\continuous\IndexWorker\App\_Data\Config
- o \App\_Data\jobs\continuous\IndexWorker\App\_Data\Diagnostics
- o \App\_Data\jobs\continuous\IndexWorker\App\_Data\Models

2. Delete all the files but *NOT* the folders in the following folders.

Leave the folders themselves to ensure that the file permissions are preserved:

- o \App\_Data\jobs\continuous\AutomationEngine
- o \App\_Data\jobs\continuous\IndexWorker

3. 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

4. Copy all the files and folders from the Sitecore 9.1.0 rev. 001564 (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
5. When prompted, replace any files that have the same name.

## 7.3.1 Add New Connection Strings

### Add reporting connection string to 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" />
```

### Add reporting.client connection strings to 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=  
"/>
```

## 7.3.2 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.

## 7.4 Configure the Search Provider

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

### 7.4.1 Configure Solr

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.

To configure the new Solr cores:

1. 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 `mysolrinstance`
- o `mysolrinstance_rebuild`

#### Important

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

2. Use the [Solr Schema API](#) to apply the schema changes from the `xconnect\App_data\solrcommands\schema.json` file for both cores.

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

```
$pathToXdbSchemaFile="<path_to_instance>\xconnect\App_data\solrcommands\schema.json"
$apiUrl="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
```

3. 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/xdb" />
```

### 7.4.2 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 9.1.0, 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\AppData\jobs\continuous\IndexWorker\AppData\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`

## 7.5 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\AppData\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\AppData\Config\Sitecore\Collection\sc.Xdb.Collection.Data.Sql.KeyStoreProviders.xml` file, replace `<ClientIdAppSettingsKey>` and `<ClientSecretAppSettingsKey>` values with the appropriate values from the `xconnect\AppData\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\AppData\Config\AppSettings.config` file.

3. Disable the `xconnect\AppData\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](#).



## 7.6 Install the Sitecore Cortex™ Processing Engine

The Sitecore Cortex™ Processing Engine is a new role in Sitecore XP 9.1.0. You can use it with or without Microsoft Machine Learning Server (MLS).

To install the Sitecore Cortex™ Processing Engine:

1. In the `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` should point to the new Processing Engine role.

2. In the `xconnect\App_data\jobs\continuous\ProcessingEngine\App_Data` folder, add a valid `license.xml` file
3. 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.
4. 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
tcore.ProcessingEngine.exe -service" start= auto obj= "NT AUTHORITY\LOCAL SERVICE"
DisplayName= "Sitecore Processing Engine - instanceName.xconnect-
ProcessingEngineService"
```

### 7.6.1 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

By default, there are no worker processes in the Sitecore Cortex™ Processing Engine that use the MLS integration.

To install Microsoft Machine Learning Server:

1. Install [Microsoft Machine Learning Server](#)
2. Navigate to `C:\Program Files\Microsoft\ML Server\R_SERVER\bin\x64\Rgui.exe` and run *RGui* as an administrator.
3. In *RGui*, run the following commands:

```
```\nlib <- tail(.libPaths(), n=1)\nrepo <- 'https://cran.microsoft.com/snapshot/2017-07-04';\ninstall.packages('openssl', lib, repos=repo)\ninstall.packages('curl', lib, repos=repo)\ninstall.packages('httr', lib, repos=repo)\n```\n
```

4. To configure Machine Learning Server to operationalize analytics on a single machine, set up the *web* and *compute* nodes.

For more information about configuring MLS, see [Microsoft's documentation](#).

5. Configure the HTTPS protocol for the *web* node.

For more information about how to configure HTTPS, see [Microsoft's documentation](#).

6. Make a note of the *web* node (port 12800 by default), username, and password, For example:

```
https://admin:Secret123@localhost:12800/
```

If you want to use special characters in the `processing.engine.mrs` connection string, the characters must be encoded, for example, `Secret123#` becomes `Secret123%23`.

To configure MLS to integrate with the Sitecore Cortex™ Processing Engine:

## Note

If the Sitecore Cortex™ Processing Storage services are not publicly available and you have installed Microsoft Machine Learning Server on a separate machine, you must update the `C:\Windows\System32\drivers\etc\hosts` file and map the IP address to the host name of the Sitecore Cortex Processing web services instance.

1. Create a folder to store the certificates that Machine Learning Server needs, for example:  

```
c:\temp\certificates
```
2. Open the **Control Panel** and search for *Manage computer certificates*.
3. To export the xConnect client certificate for the hosting *Sitecore Cortex™ Processing Blob Storage* web endpoint, in the **certlm** dialog box, expand *Personal, Certificates*.
4. Right-click the certificate for your installation, for example, `scg1.xconnect_client` and then click **All Tasks, Export...**
5. The **Certificate Export Wizard** opens and guides you through the process.
6. In the wizard, select the following options:
  - o *Yes, export the private key*
  - o *Export all extended properties*
  - o *Password-protect the file - for example, secret.*
7. Save the file with any name in the folder that you created earlier – for example:

```
c:\temp\certificates\client.pfx.
```

8. If the *Sitecore Cortex™ Processing Table* web endpoint is located on a different instance, you must perform steps 7-13 to export the certificate with the private key for the xConnect instance that hosts the *Sitecore Cortex™ Processing Table* web endpoint.
9. To export the certificate for the certification authority that issued the certificates for the *Sitecore Cortex™ Processing Blob Storage* and *Sitecore Cortex™ Processing Table* web endpoints, in the **certlm** dialog box, expand *Trusted Root Certification Authorities, Certificates*.
10. Right-click the certificate for your installation, for example: *DO\_NOT\_TRUST\_SitecoreRootCert* and then click **All Tasks, Export...**
11. The **Certificate Export Wizard** opens and guides you through the process.
12. In the wizard, select the following options:
  - *No, do not export the private key*
  - *Base-64 encoded X.509 (.cer)*
13. Save the file with any name in the folder that you created earlier, for example:

```
c:\temp\certificates\auth.cer
```

14. In the `xconnect\App_data\jobs\continuous\ProcessingEngine\App_Config\ConnectionStrings.config` file, update the connection strings for the *Sitecore Cortex™ Processing Engine* Windows service:

```
<add name="processing.engine.mrs"
connectionString="https://admin:Secret123@localhost:12800/" />

<add name="processing.webapi.blob.mrs.certificate" connectionString="Path=
c:\\temp\\certificates\\client.pfx.;Password=secret" />

<add name="processing.webapi.certificate.authority"
connectionString="c:\\temp\\certificates\\auth.cer" />

<add name="processing.webapi.table.mrs.certificate" connectionString="Path=
c:\\temp\\certificates\\client.pfx;Password=secret" />
```

## 7.7 Resume the Windows Services

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

## Chapter 8

### Post-upgrade Steps

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

This chapter contains the following sections:

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

## 8.1 Configure the Sitecore Identity Client

The Sitecore instance is a default client for Sitecore identity.

After the upgrade, you must configure several parameters:

1. To register the Sitecore Identity server on Sitecore, 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. In 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" />
```

## 8.2 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**.
- 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](#).

### 8.2.1 Cache Configuration

This section is *only* relevant if you have upgraded from any version of Sitecore XP 8.1 to Sitecore XP 9.1.0.

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

For more information, see the topic [cache API changes](#) topic on the Sitecore Documentation Site.

For more information about cache configuration, download the [Cache Configuration Reference](#) document.

## 8.3 xDB Maintenance

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

### 8.3.1 Enable xDB and Experience Analytics

You must now enable the Sitecore Experience Database (xDB) and the Experience Analytics functionality that you disabled in the *Disable Experience Analytics* 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`

### 8.3.2 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](#).



## 8.3.3 Redeploy the Marketing Definitions

To redeploy the marketing definitions:

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

### Note

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

## 8.3.4 Upgrade the Session Database

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

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

## 8.4 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.

### 8.4.1 Delete Unwanted Marketing Definitions and Marketing Taxonomies

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

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

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

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

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

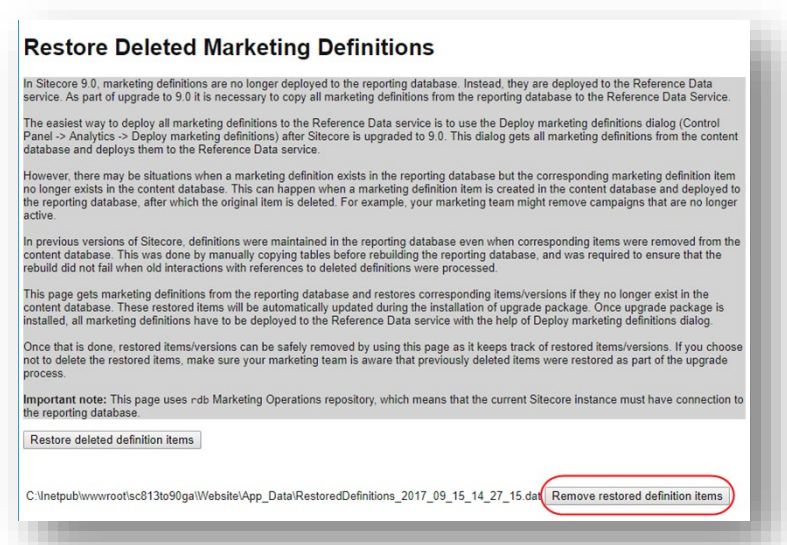
- In a text editor, in the `\App_Data` folder, open the `RestoredTaxonomies_{timestamp}.dat` file.

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

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

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

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

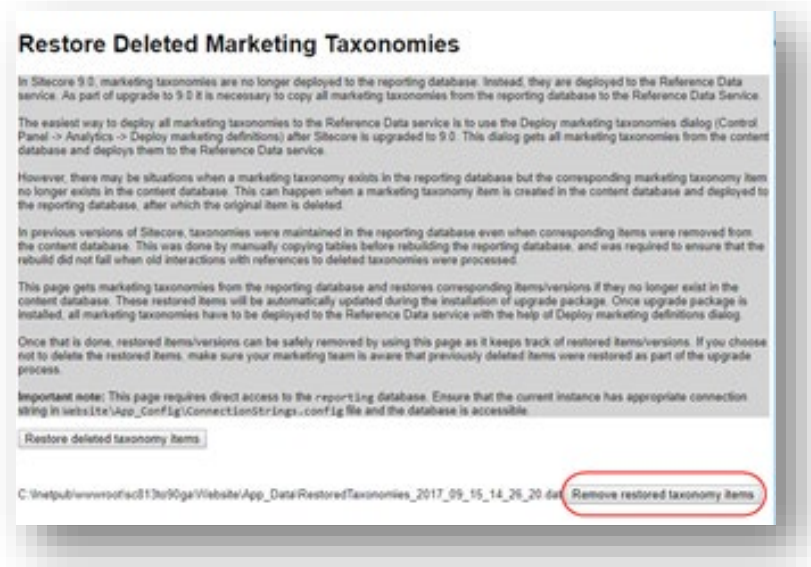


To delete unwanted marketing taxonomy items from the *Master* database:

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**.



## 8.4.2 Update Customizations for Reporting

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

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

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

## 8.4.3 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](#).

### 8.4.4 Upgrade Path Analyzer

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

#### Update Custom Maps

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

#### Redeploy the Path Analyzer Maps

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

To redeploy the Path Analyzer maps:

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

## 8.5 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 the tasks described in this section.

### 8.5.1 Upgrade the Contact Merge Data

In Sitecore XP 9.0 Update 2, the contact merge behavior was changed. In Sitecore XP 9.1.0, 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.

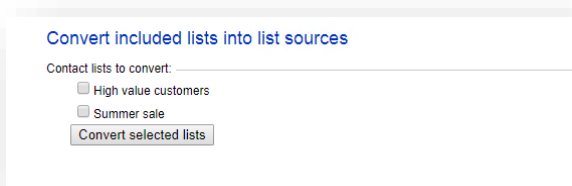
If you followed our security recommendation and [restricted access to the client](#) from the processing server role, you must enable it to access the `ContactsMerge` page. When the upgrade process is finished, you can restrict access again.

### 8.5.2 Upgrade the Included Lists in List Manager

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

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

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



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

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

## 8.5.3 Upgrade the Experience Analytics Data

After you upgrade to Sitecore XP 9.1.0, you must upgrade the Experience Analytics data.

To upgrade the Experience Analytics data in Sitecore XP 9.1.0, you need the following file:

- `SXP_reporting_migrate_experience_analytics_data.sql`

This file is available in the `Database Upgrade Script.zip` file.

To upgrade the Experience Analytics data in Sitecore XP 9.1.0:

1. In SQL Server, select the *Reporting* database in Sitecore XP 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 9

### 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

## 9.1 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 9.1.0* 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 10

### Migrate xDB data

**Important**

If you are upgrading from Sitecore XP 8.1 – Sitecore XP 8.2 Update 7 to Sitecore XP 9.1.0, you must perform this step.

If you are upgrading from Sitecore XP 9.0 or later to Sitecore XP 9.1.0, 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

## 10.1 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 9.1.0 rev. 001564 (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" />
```

### 10.1.1 Migrate the xDB Data

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

You can download the xDB Data Migration tool from the [Sitecore XP 9.0 Downloads page](#).

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

## Chapter 11

### 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

## 11.1 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

### 11.1.1 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.

### 11.1.2 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 12

### Using the Update Center

If you are running Sitecore XP 9.0.2, you can use the Update Center to upgrade to Sitecore XP 9.1.0.

Before you can use the Update Center to upgrade to Sitecore XP 9.1.0, 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 9.1.0
- Troubleshooting

## 12.1 Preparing for the Upgrade

### 12.1.1 Steps for Upgrading xDB Functionality

#### Stop Two Windows Services

You must stop these Windows services:

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

#### 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_dispatchqueue.sql` script for the *EXM.Master* database.

### 12.1.2 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*.

### 12.1.3 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.

### 12.1.4 Add the Security Connection String

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

Before you upgrade to Sitecore XP 9.1.0, you must add the *security* connection string.

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

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

## 12.2 Using the Update Center to Upgrade to Sitecore XP 9.1.0

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

### 12.2.1 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. Install the `SitecoreUpdate Installation Wizard 4.0.0 rev.181002.zip` package.

#### Note

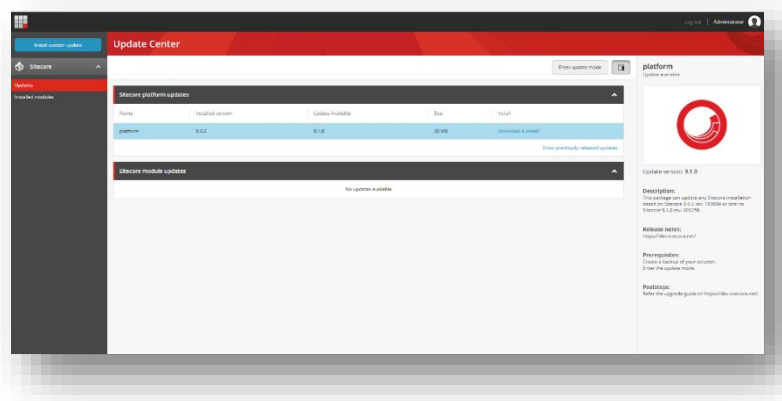
Ensure that Internet Information Services (IIS) is configured to allow access to the `\sitecore\admin` folder.

### 12.2.2 Install the Upgrade Package

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

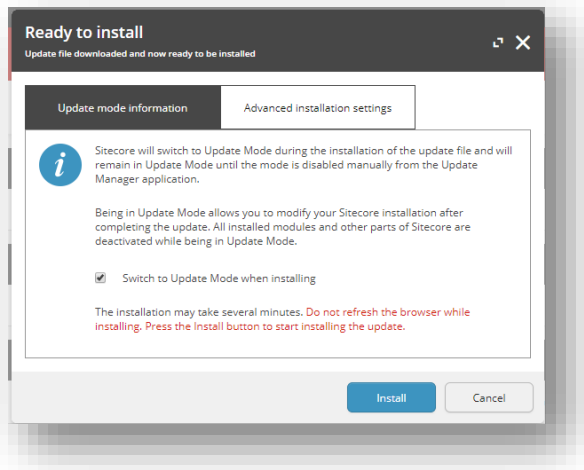
To install the upgrade package:

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



2. Select the Sitecore XP 9.1.0 upgrade package and then click **Download and Install**

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



4. Wait until the package is installed.

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**.

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

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

Review the configuration files changes:

- Familiarize yourself with new Helix folder structure that is used with Sitecore XP 9.1.0.
- 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.

6. Update Solr.

7. Upgrade xConnect.

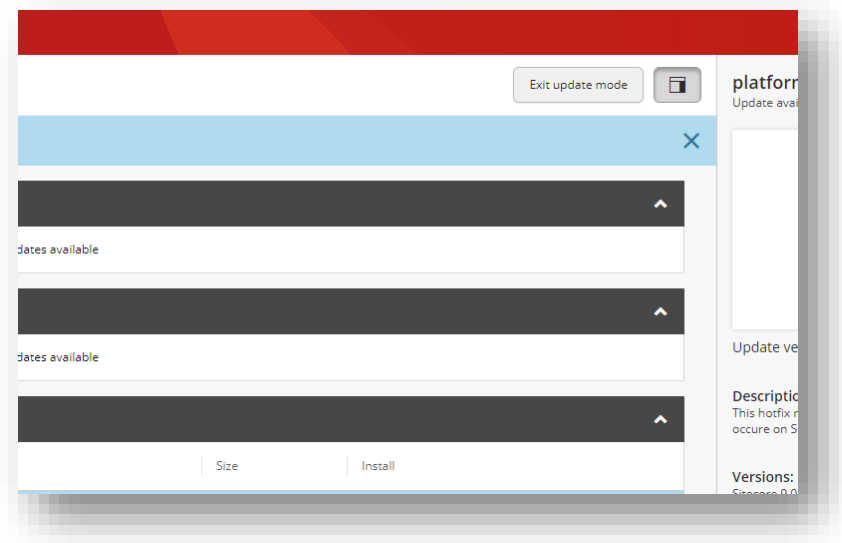
If you used xConnect in Sitecore XP 9.0.2, you must upgrade it.

For more information about upgrading xConnect, see the chapter *Upgrade xConnect to Sitecore XP 9.1.0*.

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



9. In the Update Center, click **Exit update mode**.



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

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

## 12.3 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 9.1.0 in the Update Center

Check the error message that appears in the Update Center.

### When I click Install, the installation fails

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

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

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

The update process is divided into three steps:

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

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

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

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

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

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

### My Sitecore instance is unstable after being updated

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

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

Ensure that you have setup any additional services that are required for your Sitecore XP 9.1.0 instance to run.

## Chapter 13

### Appendix A: The xConnect Application Preparation

#### Script

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

To use the script:

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

```

:: 7.3 Step 1
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

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

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

:: 7.3 Step 2
del App_Data\jobs\continuous\AutomationEngine /q
del App_Data\jobs\continuous\IndexWorker /q
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

:: 7.3 Step 3
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.