

# Sitecore Experience Platform Upgrade Guide

Sitecore Experience Platform 9.0 Update 2

*Upgrade to Sitecore XP 9.0 Update 2 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
2.2	Files and Scripts for Upgrading from Sitecore XP 8.1 or Later .....	7
2.3	Microsoft Visual C++ 2015 Redistributable Requirements .....	9
Chapter 3	Prepare for the Upgrade .....	10
3.1	Preparing to Upgrade from Sitecore XP 8.1 – Sitecore XP 8.2 Update 7 .....	11
3.1.1	Restore Deleted Marketing Taxonomies and Marketing Definitions .....	11
3.1.2	Validate the Names of the Marketing Definition Items .....	12
3.1.3	Disable the Email Experience Manager Module .....	12
3.1.4	Upgrade the Databases .....	12
3.1.5	Deploy the New Databases .....	12
3.2	Preparing to Upgrade from Sitecore XP 9.0 – Sitecore XP 9.0.1 .....	14
3.2.1	Disable Experience Analytics .....	14
3.2.2	Stop Two Windows Services .....	14
3.2.3	Disable the Data Migration Tool .....	14
3.2.4	Upgrade the Databases .....	14
3.3	Final Preparations for Upgrading from All Versions .....	15
3.3.1	Content Testing .....	15
3.3.2	Disable the Web Forms for Marketers Module .....	15
3.3.3	Enable Email Experience Manager .....	15
3.3.4	Disable the Sitecore Experience Database .....	16
3.3.5	Indexing and Performance .....	17
Chapter 4	Install the Upgrade Package .....	18
4.1	Use the Update Installation Wizard to Install the Upgrade Package .....	19
4.2	The Upgrade Package Analysis .....	21
4.2.1	Resolve Breaking Changes in Custom Code .....	21
4.2.2	Resolve Configuration File Conflicts .....	22
Chapter 5	Configure Sitecore .....	26
5.1	Review Custom Changes in Configuration Files .....	27
5.2	Specify the Server Role and the Search Provider .....	28
5.3	Remove Deprecated Indexes .....	29
5.4	Update Solr .....	30
5.4.1	Specify the Solr connecting string .....	30
5.4.2	Configure Managed Schema .....	30
5.4.3	Populate the Solr Managed Schema .....	30
Chapter 6	Upgrade Web Forms for Marketers .....	32
6.1	Web Forms for Marketers and Sitecore Forms .....	33
6.1.1	Upgrade the Web Forms for Marketers Module .....	33
Chapter 7	Install xConnect .....	34
7.1	Install and Configure Sitecore xConnect .....	35
Chapter 8	Upgrade xConnect from Sitecore XP 9.0 Initial Release to Sitecore XP 9.0 Update 2 .....	36
8.1	Prepare for the Upgrade .....	37
8.1.1	Prerequisites for Upgrading xConnect .....	37
8.1.2	Upgrade the Databases .....	37
8.1.3	Configure Solr .....	38
8.2	Configuration Changes .....	39
8.2.1	Upgrade of the Sitecore XConnect Search Indexer service .....	45
8.2.2	Upgrade the Sitecore Marketing Automation Engine service .....	48
8.2.3	Resuming Services .....	51
Chapter 9	Upgrade xConnect from Sitecore XP 9.0.1 to Sitecore XP 9.0 Update 2 .....	52
9.1	Prepare for the Upgrade .....	53
9.1.1	Prerequisites for Upgrading xConnect .....	53

# Sitecore Experience Platform Upgrade Guide

9.1.2	Upgrade the Databases.....	53
9.2	Configuration Changes.....	54
9.2.1	Upgrade of the Sitecore XConnect Search Indexer service .....	57
9.2.2	Upgrade the Sitecore Marketing Automation Engine service .....	59
9.2.3	Resuming Services .....	59
Chapter 10	Post-upgrade Steps .....	60
10.1	General Maintenance .....	61
10.1.1	Cache Configuration .....	61
10.2	xDB Maintenance .....	62
10.2.1	Enable xDB and Experience Analytics .....	62
10.2.2	Using EXM .....	62
10.2.3	Redeploy the Marketing Definitions.....	62
10.2.4	Upgrade the Session Database .....	63
10.3	Upgrading from Sitecore XP 8.1 – Sitecore XP 8.2.7.....	64
10.3.1	Delete Unwanted Marketing Definitions and Marketing Taxonomies.....	64
10.3.2	Update Customizations for Reporting.....	65
10.3.3	Upgrade Experience Analytics.....	65
10.3.4	Upgrade Path Analyzer .....	66
10.4	Upgrading from Sitecore XP 9.0 or Sitecore 9.0 Update 1.....	67
10.4.1	Upgrade the Contact Merge Data .....	67
Chapter 11	Upgrade Multiple Instances.....	68
11.1	Upgrade a Scaled Environment With Multiple Instances.....	69
Chapter 12	Migrate xDB data.....	70
12.1	Use the xDB Data Migration Tool to Migrate xDB data.....	71
12.1.1	Migrate the xDB Data .....	71
Chapter 13	Breaking Changes from EXM 3.5 .....	72
13.1	Breaking Changes .....	73
13.1.1	Factories Replaced.....	73
13.1.2	Client API and Messaging Bus .....	73

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 9.0 rev. 180604 Update 2.

If you are upgrading from Sitecore XP 8.1 rev. 151003 (Initial Release) or later, you can upgrade directly to Sitecore XP 9.0 rev. 180604 Update 2.

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 9.0 Update 2.

This guide contains the following chapters:

- **Introduction**
- **Prerequisites**
- **Prepare for the Upgrade**
- **Install the Upgrade Package**
- **Configure Sitecore**
- **Upgrade Web Forms for Marketers**
- **Install xConnect**
- **Upgrade xConnect from Sitecore XP 9.0 Initial Release to Sitecore XP 9.0 Update 2**
- **Upgrade xConnect from Sitecore XP 9.0.1 to Sitecore XP 9.0 Update 2**
- **Post-upgrade Steps**
- **Upgrade Multiple Instances**
- **Migrate xDB data**
- **Breaking Changes from EXM 3.5**

## Chapter 2

### Prerequisites

This chapter describes the prerequisites required to upgrade to Sitecore XP 9.0 rev. 180604 Update 2.

This chapter contains the following section:

- Prerequisites
- Files and Scripts for Upgrading from Sitecore XP 8.1 or Later
- Microsoft Visual C++ 2015 Redistributable Requirements

## 2.1 Prerequisites

Before you upgrade to this version of Sitecore XP9, 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 Installation Guide.

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

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

- An instance of [Sitecore 8.1 rev. 151003 \(Initial Release\) or later](#)
- Sitecore 9.0.2 rev. 180604 (update package).zip

This file contains the following files:

- Sitecore 9.0.2 rev. 180604 (update package for 8.1).zip
- Sitecore 9.0.2 rev. 180604 (update package for 8.2).zip
- Sitecore 9.0.2 rev. 180604 (update package for 9.x).zip

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

- Sitecore 9.0.2 rev180604 (update package for 9.x).update

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

- Sitecore Update Installation Wizard 3.1.2 rev. 180406.zip
- Sitecore 9.0.2 rev. 180604 (upgrade files).zip

This file contains:

- Database Upgrade Script.zip

This file contains the following database upgrade scripts:

- CMS9.0\_BeforeInstall.sql
- CMS90\_db\_core\_BeforeInstall.sql
- SXP90\_BeforeInstall.sql
- SXP90\_db\_marketingautomation\_BeforeInstall.sql
- SXP90\_db\_processing\_pools\_BeforeInstall.sql
- SXP90\_db\_processing\_tasks\_BeforeInstall.sql
- SXP90\_db\_collection\_BeforeInstall.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 to support new functionality or to increase 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 Experience Platform Upgrade Guide

- `Sitecore.Experienceforms.dacpac`
- `Sitecore.Messaging.dacpac`
- `Sitecore.Processing.tasks.dacpac`
- [xDB Data Migration Tool](#) – required if you have existing analytics data in MongoDB that you want to use with Sitecore XP 9.0 Update 2.

## Note

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

## 2.3 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.*

## Chapter 3

### Prepare for the Upgrade

Before you upgrade to Sitecore XP 9.0 Update 2 rev. 180604, 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 both upgrade paths.

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

This chapter contains the following sections:

- Preparing to Upgrade from Sitecore XP 8.1 – Sitecore XP 8.2 Update 7
- Preparing to Upgrade from Sitecore XP 9.0 – Sitecore XP 9.0.1
- Final Preparations for Upgrading from All Versions

## 3.1 Preparing 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 9.0 Update 2, you must perform all the steps described in this section.

If you are upgrading from Sitecore XP 9.0 or Sitecore XP 9.0.1 to Sitecore XP 9.0 Update 2, you do not have to perform all the steps described in this section.

### 3.1.1 Restore Deleted Marketing Taxonomies and Marketing Definitions

If you are using the Sitecore Experience Database (xDB) functionality, (when 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.

To restore deleted marketing taxonomy items:

1. Unpack the `MarketingDefinitionsUpgrade.zip` file.
2. Copy the `RestoreDeletedMarketingTaxonomies.aspx` page to the `\sitecore\admin` folder of your website.
3. To open the **Restore Deleted Marketing Taxonomies** page, enter 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.

4. Click **Restore deleted taxonomy items**.

At this point, 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. Unpack the `MarketingDefinitionsUpgrade.zip` file.
2. Copy the `RestoreDeletedMarketingDefinitions.aspx` file to the `sitecore/admin` folder of your website.
3. 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.

4. Click **Restore deleted definition items**.

At this point, 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.

## 3.1.2 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. To open the **Definition Items Name Validator** page, enter 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 any 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.

## 3.1.3 Disable the Email Experience Manager Module

To disable the Email Experience Manager module:

1. Delete the following folders:
  - o `\App_Config\Include\EmailExperience`
  - o `\App_Config\Include\Z.EmailExperience`
2. Delete the following file:
  - o `\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`.

## 3.1.4 Upgrade the Databases

To upgrade the databases:

1. Execute the `CMS9.0_BeforeInstall.sql` script for the *Core*, *Master*, and *Web* databases.
2. Execute the `CMS90_db_core_BeforeInstall.sql` script for the *Core* database only.
3. If you are using the xDB functionality, you must execute the `SXP90_BeforeInstall.sql` script for the *Reporting* database.

## 3.1.5 Deploy the New Databases

In Sitecore 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 `\Databases` folder, in the `Sitecore 9.0.2 rev. 180604 (upgrade files).zip` package.

# Sitecore Experience Platform Upgrade Guide

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;DataSource=(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;DataSource=(server);Database=Sitecore.Processing.Tasks" />
```

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

## 3.2 Preparing to Upgrade from Sitecore XP 9.0 – Sitecore XP 9.0.1

### Important

If you are upgrading from Sitecore XP 8.1 – Sitecore 8.2 Update 7 to Sitecore XP 9.0 Update 2, you do not have to perform all the steps described in this section.

If you are upgrading from Sitecore XP 9.0 or Sitecore 9.0.1 to Sitecore XP 9.0 Update 2, you must perform all the steps described in this section.

### 3.2.1 Disable Experience Analytics

To disable Experience Analytics:

1. Unpack the `Sitecore 9.0.2 rev. 180604 (upgrade files).zip` file and locate the `Disable.xAnalytics.config` file.
2. In the `\App_Config\Include` folder, create a new `\Z.Custom` folder (if don't have it already) and copy the `Disable.xAnalytics.config` file to the `\App_Config\Include\Z.Custom` folder.

This prevents item saved events from occurring during the upgrade process.

### Important

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

### 3.2.2 Stop Two Windows Services

You must stop these Windows services:

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

### 3.2.3 Disable the 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 9.0 Update 2. After you have migrated your data, you no longer need the tool.

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

### 3.2.4 Upgrade the Databases

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

1. Execute the `SXP90_db_processing_tasks_BeforeInstall.sql` script for the `Processing.tasks` database.
2. If you are using the xDB functionality, you must execute the `SXP90_BeforeInstall.sql` script for the `Reporting` database.

## 3.3 Final Preparations for Upgrading from All Versions

If you are upgrading from Sitecore XP 8.1 – Sitecore XP 9.0.1, you must perform the following steps.

### 3.3.1 Content Testing

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.

### 3.3.2 Disable the Web Forms for Marketers Module

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`

#### Important

After the upgrade of Sitecore Experience Platform is completed, you must also upgrade the modules. For more information, see the section *Upgrade Web Forms for Marketers* in this guide.

### 3.3.3 Enable Email Experience Manager

#### Important

If you are upgrading from Sitecore XP 8.1 – Sitecore XP 9.0 Initial Release to Sitecore XP 9.0 Update 2, you must perform the steps described in this section.

If you are upgrading from Sitecore XP 9.0 Update 1 to Sitecore XP 9.0 Update 2, you do not have to perform the steps described in this section.

## Upgrade from Sitecore XP 8.1.0 – Sitecore XP 8.2 Update 7

To use Email Experience Manager (EXM), you must add the new connection strings.

To enable EXM:

1. In the `\App_Config` folder, in the `ConnectionStrings.config` file, add the connection string for the `Sitecore.Exm.master` database.

For example:

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

2. Add the following three connection strings:

- `EXM.InternalApiKey`

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

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

# Sitecore Experience Platform Upgrade Guide

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" />
```

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

## Upgrade from Sitecore XP 9.0 Initial Release

To use Email Experience Manager (EXM), you must deploy the new databases and add the connection strings.

To enable EXM:

1. Deploy the `Sitecore.Exm.master.dacpac` and the `Sitecore.Messaging.dacpac` files to SQL Server
2. In the `\App_Config` folder, in the `ConnectionStrings.config` file, add the connection string for the `Sitecore.messaging` database.

For example:

```
<add name="messaging" connectionString="user id=user;password=password;DataSource=(server);Database=Sitecore.Messaging" />
```

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

3. You must create a SQL login with the appropriate access rights to the `Messaging` database. This SQL login must be a member of the `db_datareader` and `db_datawriter` security roles.

To create this login, run the following script:

```
: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
```

### 3.3.4 Disable the Sitecore Experience Database

To disable xDB:

1. Unpack the `Sitecore 9.0.2 rev. 180604 (upgrade files).zip` file and the `Disable xDB.zip` file.
2. In the `\App_Config\Include` folder, create a new `\Z.Custom` folder and copy the `Disable.xDB.config` file to the `\App_Config\Include\Z.Custom` 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.

## 3.3.5 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, you can disable indexing by removing the following processor from the `Sitecore.ContentSearch.config` file:

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

## Chapter 4

### Install the Upgrade Package

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

**Note**

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

This chapter contains the following sections:

- Use the Update Installation Wizard to install the upgrade package
- The upgrade package analysis

## 4.1 Use the Update Installation Wizard to 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 3.1.2 rev.180406.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:

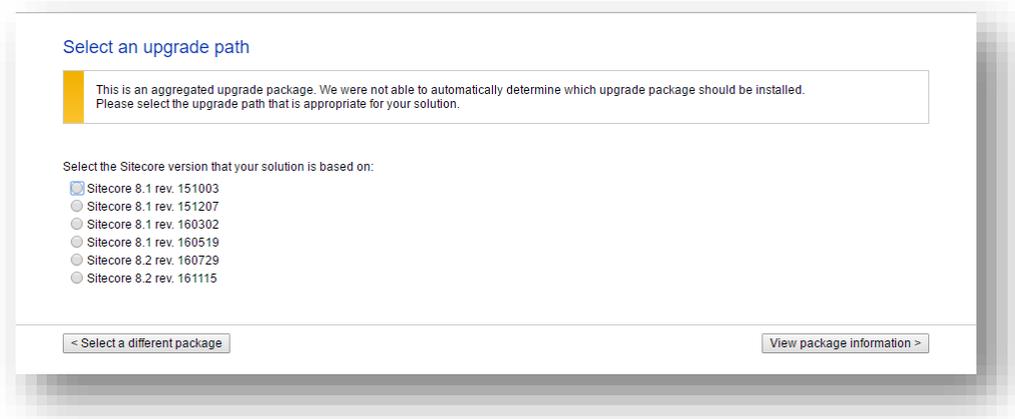
```
http://<hostname>/sitecore/admin/UpdateInstallationWizard.aspx
```

### Note

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

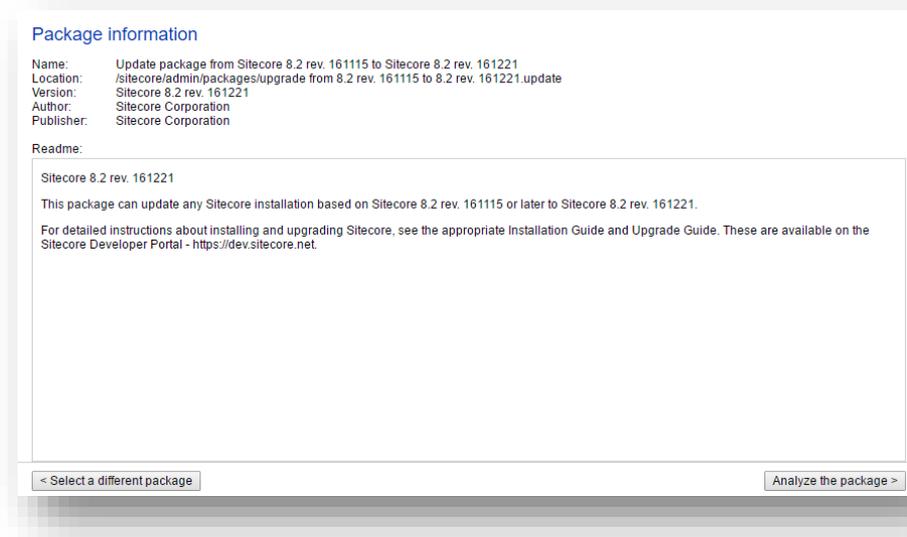
4. On the **Welcome to Sitecore update installation wizard** page, click **Select a package**.
5. On the **Select a package** page, click **Choose File** and navigate to the folder where you saved the update file for Sitecore 9.0.2 rev. 180604. Select the update file for the Sitecore version that you 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**.

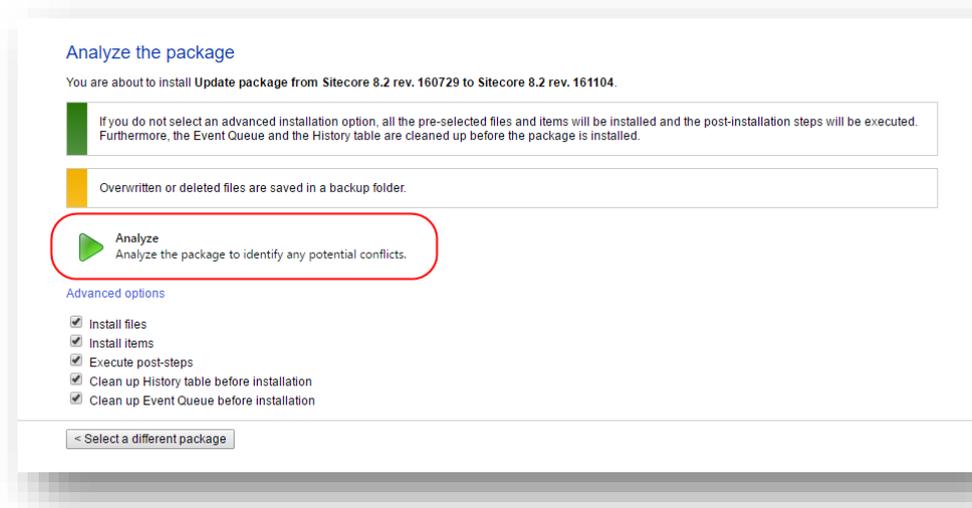


# Sitecore Experience Platform Upgrade Guide

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 section *Configure*.

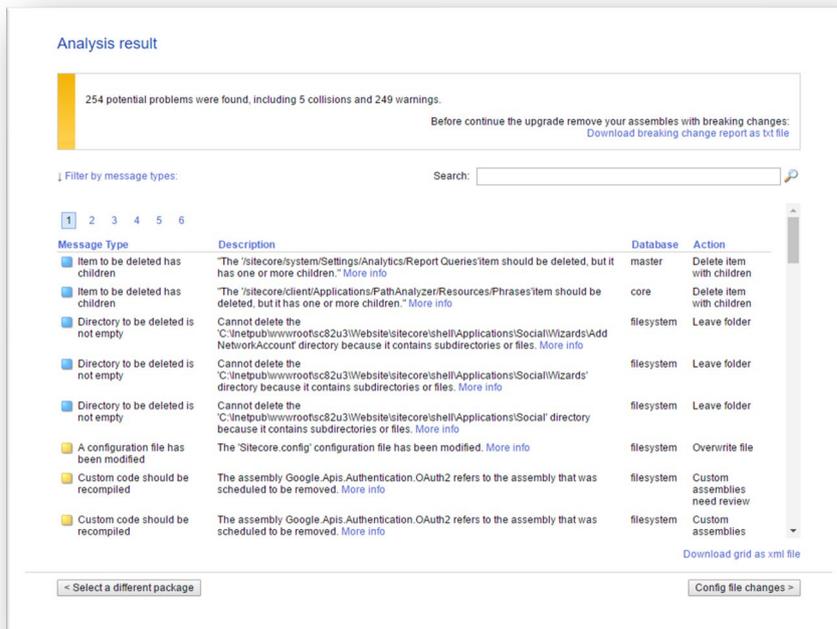
## 4.2 The Upgrade Package Analysis

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.



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

### Important

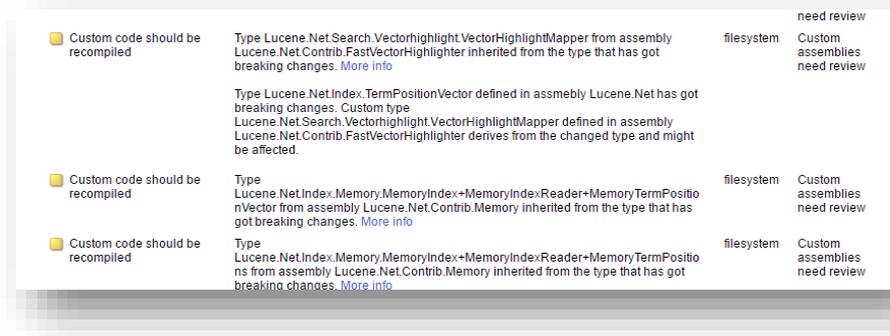
While the majority of the actions ensure the best upgrade experience, if the analysis identifies custom code that must be recompiled or configuration file conflicts that must be resolved, you must address these manually.

### 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**.



Message Type	Description	Database	Action
Custom code should be recompiled	Type Lucene.Net.Search.Vectorhighlight.VectorHighlightMapper from assembly Lucene.Net.Contrib.FastVectorHighlighter inherited from the type that has got breaking changes. <a href="#">More info</a>	filesystem	Custom assemblies need review
Custom code should be recompiled	Type Lucene.Net.Index.TermPositionVector defined in assmebly Lucene.Net has got breaking changes. Custom type Lucene.Net.Search.Vectorhighlight.VectorHighlightMapper defined in assembly Lucene.Net.Contrib.FastVectorHighlighter derives from the changed type and might be affected.	filesystem	Custom assemblies need review
Custom code should be recompiled	Type Lucene.Net.Index.Memory.MemoryIndex+MemoryIndexReader+MemoryTermPositionVector from assembly Lucene.Net.Contrib.Memory inherited from the type that has got breaking changes. <a href="#">More info</a>	filesystem	Custom assemblies need review

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

To resolve these warnings:

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

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

Or

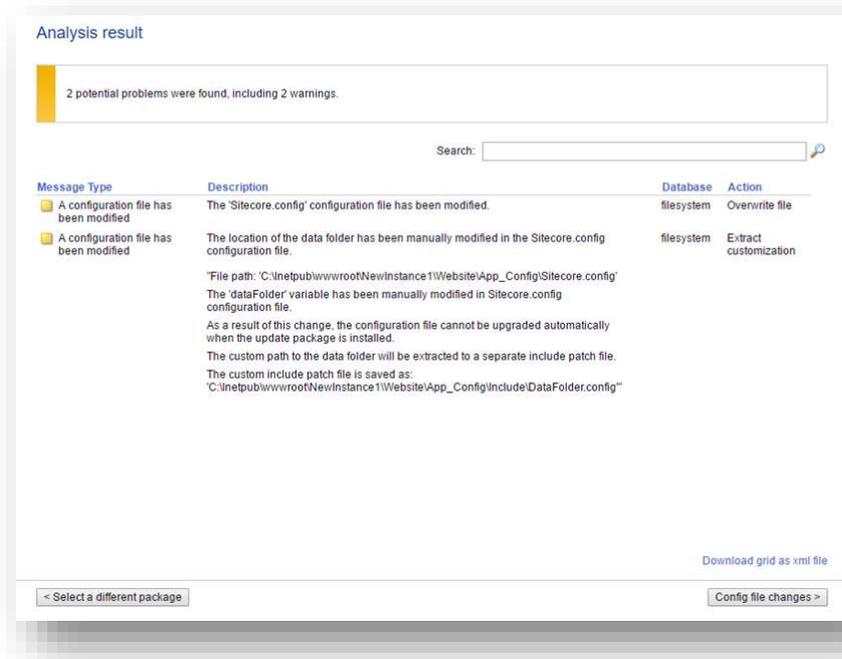
- If you are sure that the affected code is not used during Sitecore startup or in important Sitecore operations such as item management, security management, file management, or in regular website requests that require Sitecore functionality, you can ignore the warnings.

## Note

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

## 4.2.2 Resolve Configuration File Conflicts

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



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

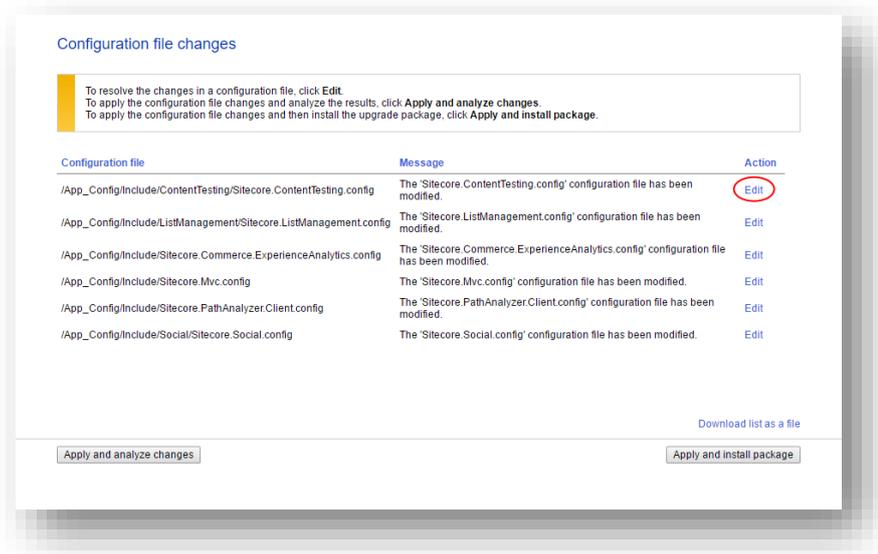
# Sitecore Experience Platform Upgrade Guide

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

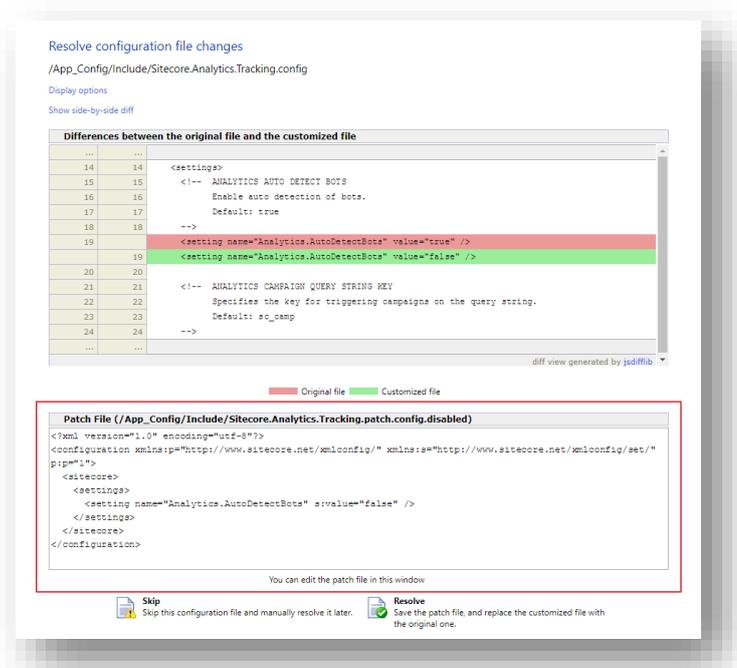
To see more information about a configuration file change, click the description.

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**.

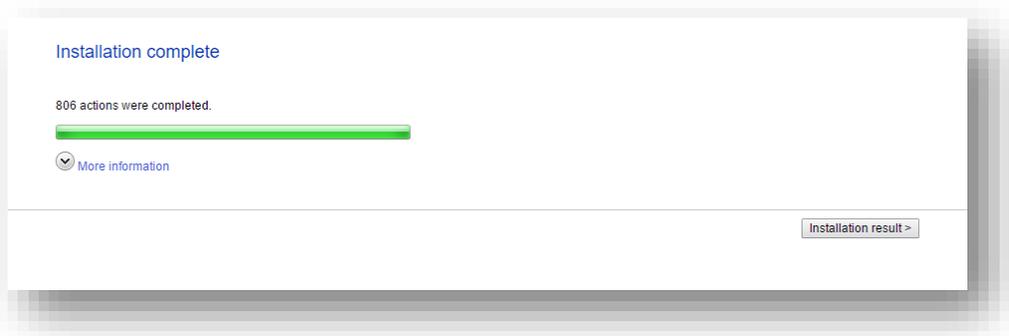
## Note

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.

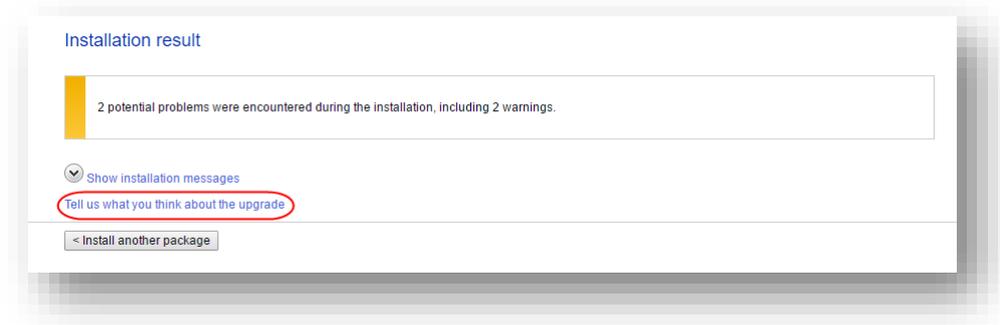
10. When the update process is finished, on the **Installation complete** page, click **Installation result**.



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 reviewing the custom changes in the configuration files and perform the procedures described in this chapter.

This chapter contains the following sections:

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

## 5.1 Review Custom Changes in Configuration Files

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 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.

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

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

To enable your customizations:

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

### Note

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 9.0 Update 2, 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.

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

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="Solr" />
```

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 9.0 Update 2, you must perform this step.

If you are upgrading from Sitecore XP 9.0 or Sitecore 9.0.1 to Sitecore XP 9.0 Update 2, 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 listed at the beginning of this section.

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 listed at the beginning of this section.

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 index(es) and then click **Delete**.

## 5.4 Update Solr

To continue using the Solr search provider in Sitecore 9.0 Update 2, you must update your existing Solr instance to version 6.6.2. 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 9.0 Update-2, 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="[YOUR_CONNECTION_STRING]" />
```

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 Sitecore 9.0.1 to Sitecore XP 9.0 Update 2, 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-6.6.2\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.

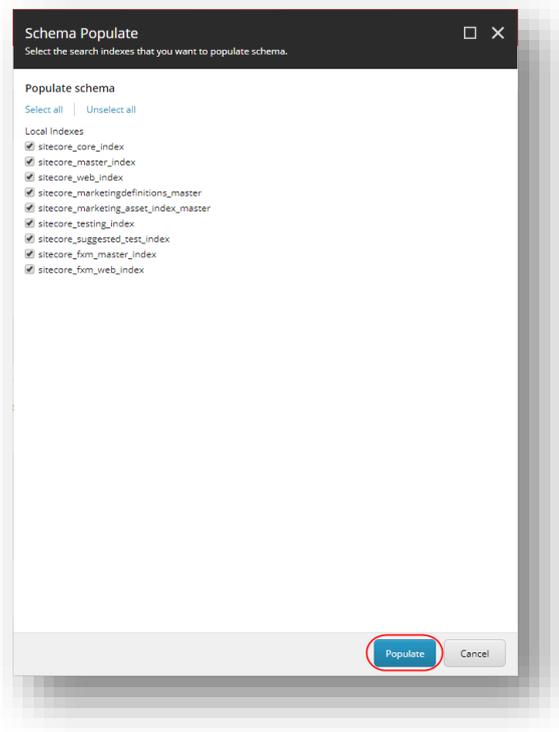
For more information, see the section *Review Custom Changes in Configuration Files*.

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

### Upgrade Web Forms for Marketers

This chapter describes how to upgrade the Web Forms for Marketers module.

For information on how to upgrade other modules, refer to the upgrade documentation for the specific module. You can download all upgrade and installation documentation from the [Sitecore Downloads page](#).

This chapter contains the following section:

- Web Forms for Marketers and Sitecore Forms

## 6.1 Web Forms for Marketers and Sitecore Forms

Sitecore 9.0 includes the new Sitecore Forms application. Digital marketers can use this application to create web forms that record and report the information provided by visitors to your website.

You cannot upgrade from WFFM to Sitecore Forms. Although WFFM is still maintained for Sitecore 9, it is not going to be further developed and we therefore recommend that you use Sitecore Forms to recreate your existing web forms and create any new forms that you need.

For more information, see the [Sitecore Forms](#) documentation.

### 6.1.1 Upgrade the Web Forms for Marketers Module

The upgrade path for WFFM is:

- WFFM 8.1.x to WFFM 8.2 Initial release
- WFFM 8.2.x to WFFM 9.0 Initial release
- WFFM 9.0.Initial release to WFFM 9.0.x

Furthermore:

- You must upgrade Sitecore XP before you upgrade WFFM.
- You must disable all the WFFM configuration files before you upgrade WFFM.

To upgrade the Web Forms for Marketers module, you must download and install the appropriate Web Forms for Marketers Update Package.

To install the update package:

1. 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>

2. The wizard guides you through the update process and helps you:
  - Upload the update package.
  - Analyze the package.
  - Install the package.

#### Note

For more information about the Update Installation Wizard, see the section *Install the Upgrade Package*.

## Chapter 7

### Install xConnect

If you want to use the Sitecore Experience Database (xDB) functionality in Sitecore 9.0 Update 2, you must install and configure Sitecore xConnect.

**Important**

If you are upgrading from Sitecore XP 8.1 – Sitecore XP 8.2 Update 7 or running Sitecore XP 9.0 or Sitecore XP 9.0.1 in CMS-only mode, you must perform the tasks described in this chapter.

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

This chapter contains the following sections:

- Install and Configure Sitecore xConnect

## 7.1 Install and Configure Sitecore xConnect

For more detailed information about installing xConnect — including deployment topologies — see the *Sitecore Experience Platform 9.0 Update 2 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"/>
```

### Note

These connection strings are for a single instance Sitecore solution only. For a scaled environment, see the [Sitecore server role configuration reference](#) documentation.

### Grant permissions to the xConnect certificates

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

To grant the read permissions:

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

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

5. Restart IIS.

## Chapter 8

# Upgrade xConnect from Sitecore XP 9.0 Initial Release to Sitecore XP 9.0 Update 2

This chapter describes how to upgrade xConnect when you upgrade from Sitecore XP 9.0 Initial release to Sitecore XP 9.0 Update 2.

If you are upgrading from Sitecore XP 9.0 with xDB enabled, you must perform all the tasks described in this chapter.

This chapter contains the following sections:

- Prepare for the Upgrade
- Configuration Changes

## 8.1 Prepare for the Upgrade

Before you upgrade your Sitecore instance, you must perform all the tasks described in this chapter.

### 8.1.1 Prerequisites for Upgrading xConnect

Before you can start the upgrade process, you must ensure that you have all the necessary packages and scripts.

To prepare for the upgrade:

1. [Download the packages for XP Single – Sitecore 9.0.2 rev. 180604 \(WDP XP0 packages\)](#) .zip – and unpack the zip file.

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

- o Sitecore 9.0.2 rev. 180604 (OnPrem)\_xp0xconnect.scwdp.zip
2. Unpack the Sitecore 9.0.2 rev. 180604 (OnPrem)\_xp0xconnect.scwdp.zip file so that the files it contains are ready for use.
  3. Unpack the Database Upgrade Script.zip file.

It contains several database upgrade scripts including:

- o SXP90\_db\_marketingautomation\_BeforeInstall.sql
  - o SXP90\_db\_processing\_pools\_BeforeInstall.sql
  - o SXP90\_db\_collection\_BeforeInstall.sql
4. If you are using Solr provider, ensure that you have updated Solr to version 6.6.2.
  5. If you have any functionality that accesses the contact/interaction data extraction process from outside of xConnect, you must stop it. You can restart it when the upgrade is completed. An example of this kind of functionality is rebuilding the *Reporting* database.

#### Important

You cannot just pause these tasks before the upgrade and resume them afterwards because the extraction cursors have changed in Sitecore XP 9.0.2.

### 8.1.2 Upgrade the Databases

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

#### Sitecore XP 9.0 Initial Release

To upgrade the databases:

1. Execute the SXP90\_db\_collection\_BeforeInstall.sql script for each shard of the *Collection* database.
2. Execute the SXP90\_db\_processing\_pools\_BeforeInstall.sql script for the *Xdb Processing Pools* database.
3. Execute the SXP90\_db\_marketingautomation\_BeforeInstall.sql script for the *Xdb Marketing Automation* database.
4. Ensure that you have deployed the *Messaging* database.

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

5. In the xconnect\App\_Config\ConnectionStrings.config file, add the messaging connection string:

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

**Note**

You must add the messaging connection string on every xConnect server that performs the Sitecore:XConnect:MarketingAutomation role.

## 8.1.3 Configure Solr

If you installed a new instance of Solr instead of updating the existing one to 6.6.2, you must configure new Solr cores.

To configure new Solr cores:

1. In the Solr admin console, add 2 new Solr cores that are based on the basic configset and ensure that the names follow the rule:
  - <custom\_name>, for example, "xdb".
  - <custom\_name>\_rebuild, for example, "xdb\_rebuild".

**Note**

One of the cores must be called "XXX\_rebuild".

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

Here is an example of a 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/<core_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
```

## 8.2 Configuration Changes

You must make the following configuration changes.

- In the `xconnect\App_Config\ConnectionStrings.config` file, update the `solrCore` connection string to point to the `xdb core` that is attached to Solr 6.6.2.

### Collection Service Configuration

The configuration files are stored here:

```
xconnect\App_data\config\sitecore\Collection
```

1. In the Sitecore 9.0.2 rev. 180604 (OnPrem) `_xp0xconnect.scwdp\Content\Website\App_data\config\sitecore\Collection` folder that you unpacked earlier, copy the `sc.EXM.Collection.xml` file and paste it into the `xconnect\App_data\config\sitecore\Collection` folder.

2. In the `sc.Xdb.Collection.RepositorySettings.xml` file:

- o In the `<IRepository>` section, change the `<Type>` value to:

```
<Type>Sitecore.Xdb.Collection.Repository, Sitecore.Xdb.Collection</Type>
```

- o In the `<Services>` section, after the `<IRepository>` section, add the `<IRepositoryCountersDecorator>` section:

```
<IRepositoryCountersDecorator>  
<Type>Sitecore.XConnect.Configuration.ServiceDecorator`2[[Sitecore.Xdb.Collection.IRepository, Sitecore.Xdb.Collection],  
[Sitecore.Xdb.Collection.RepositoryCountersDecorator, Sitecore.Xdb.Collection]],  
Sitecore.XConnect.Configuration</Type>  
<LifeTime>Singleton</LifeTime>  
<Options>  
<!-- Level number (0..255) that is used to sort decorators of the same service. -->  
<LevelNumber>10</LevelNumber>  
</Options>  
</IRepositoryCountersDecorator>
```

- o In the `<Services>` section, after the `<IRepositoryCountersDecorator>` section, add the `<IDataProviderCountersDecorator>` section:

```
<IDataProviderCountersDecorator>  
<Type>Sitecore.XConnect.Configuration.ServiceDecorator`2[[Sitecore.Xdb.Collection.IDataProvider, Sitecore.Xdb.Collection],  
[Sitecore.Xdb.Collection.DataProviderCountersDecorator, Sitecore.Xdb.Collection]],  
Sitecore.XConnect.Configuration</Type>  
<LifeTime>Singleton</LifeTime>  
<Options>  
<!-- Level number (0..255) that is used to sort decorators of the same service. -->  
<LevelNumber>10</LevelNumber>  
</Options>  
</IDataProviderCountersDecorator>
```

3. In the `sc.XConnect.Collection.ContentTesting.Model.Plugins.xml` file, in the `<Type>` section, change:

```
<Type>Sitecore.ContentTesting.Collection.Model.Plugins.MergeTestCombinationsHandler,  
Sitecore.ContentTesting.Collection.Model.Plugins</Type>
```

to

```
<Type>Sitecore.ContentTesting.Model.xConnect.MergeTestCombinationsHandler,  
Sitecore.ContentTesting.Model.xConnect</Type>
```

### Search Service Configuration

The configuration files are stored here:

```
xconnect\App_data\Config\Sitecore\CollectionSearch
```

# Sitecore Experience Platform Upgrade Guide

1. In the `sc.Xdb.Collection.IndexReader.AzureSearch.xml.disabled` file, in the `<Services>` section, after the `<IAzureWebClientFactory>` node, add the new `<IAzureSecondaryWebClientFactory>` service:

```
<IAzureSecondaryWebClientFactory>
<Type>Sitecore.Xdb.Collection.Search.Azure.WebClient.SecondaryCoreAzureWebClientFactory, Sitecore.Xdb.Collection.Search.Azure</Type>
<As>Sitecore.Xdb.Collection.Search.Azure.WebClient.ISecondaryWebClientFactory, Sitecore.Xdb.Collection.Search.Azure</As>
  <LifeTime>Singleton</LifeTime>
  <Options>
    <ConnectionStringName>
      <!-- Set connection string name if you need to use different connection string or different from default core name.
      collection.search.secondary-->
    </ConnectionStringName>
  </Options>
</IAzureSecondaryWebClientFactory>
```

## Core Services Configuration

The configuration files are stored here:

```
xconnect\App_data\config\sitecore\CoreServices
```

1. In the `sc.Xdb.Sql.Common.xml` file:
  - o In the `<RetryManager>` section, remove the `<Options>` section.
  - o In the `<Services>` section, before the `<RetryManager>` section, add the `<RetryManagerSettings>` section:

```
<RetryManagerSettings>
  <Type>Sitecore.Xdb.Sql.Common.ExponentialBackoffRetryManagerSettings, Sitecore.Xdb.Sql.Common</Type>
  <As>Sitecore.Xdb.Sql.Common.IRetryManagerSettings, Sitecore.Xdb.Sql.Common</As>
  <LifeTime>Transient</LifeTime>
  <Options>
    <RetryCount>10</RetryCount>
    <MinBackoff>00:00:01</MinBackoff>
    <MaxBackoff>00:00:30</MaxBackoff>
    <DeltaBackoff>00:00:10</DeltaBackoff>
  </Options>
</RetryManagerSettings>
```

## Marketing Automation Services Configuration

These configuration files are stored here:

```
xconnect\App_data\config\sitecore\MarketingAutomation
```

1. In the Sitecore 9.0.2 rev. 180604 (OnPrem) `_xp0xconnect.scwdp\Content\Website\App_data\config\sitecore\MarketingAutomation` folder that you unpacked earlier, copy the following files:

- o `sc.MarketingAutomation.PoolFilter.xml`
- o `sc.EXM.ActivityTypes.xml`

Paste them into the `xconnect\App_data\config\sitecore\MarketingAutomation` folder.

2. In the `sc.MarketingAutomation.ActivityResultHandlers.xml` file, in the `<MarketingAutomation.Factory.ActivityResultHandlerFactory>` section, change:

```
<LifeTime>Singleton</LifeTime>
```

to

```
<LifeTime>Scoped</LifeTime>
```

3. In the `sc.MarketingAutomation.Locators.xml` file, in the `<Services>` section, after the `<MarketingAutomation.Locator.MarketingOperationsPlanLocator>` section, add the new `<MarketingAutomation.Locator.MemoryPlanDefinitionCache>` section:

```
<!-- Cache that stores plan definitions in memory. -->
<MarketingAutomation.Locator.MemoryPlanDefinitionCache>
<Type>Sitecore.Xdb.MarketingAutomation.Locator.MemoryPlanDefinitionCache,
Sitecore.Xdb.MarketingAutomation</Type>
<As>Sitecore.Xdb.MarketingAutomation.Locator.IPlanDefinitionCache,
Sitecore.Xdb.MarketingAutomation</As>
<LifeTime>Singleton</LifeTime>
<Options>
  <!-- The lifetime of entries in the cache. -->
  <EntryLifetime>00:00:20</EntryLifetime>
</Options>
</MarketingAutomation.Locator.MemoryPlanDefinitionCache>
```

4. In the `sc.MarketingAutomation.PlanProcessing.xml` file:

- o In the `<Services>` section, after the `</MarketingAutomation.Processing.Results.ResultPrecedenceResultAggregator>` section, add the new `<MarketingAutomation.Enrollment.InMemoryActivityEnrollmentCache>` section:

```
<!-- A cache that stores activity enrollments in memory during a work item
processing. -->
<MarketingAutomation.Enrollment.InMemoryActivityEnrollmentCache>

<Type>Sitecore.Xdb.MarketingAutomation.Enrollment.InMemoryActivityEnrollmentCache
, Sitecore.Xdb.MarketingAutomation</Type>
<As>Sitecore.Xdb.MarketingAutomation.Enrollment.IActivityEnrollmentCache,
Sitecore.Xdb.MarketingAutomation</As>
<LifeTime>Scoped</LifeTime>
</MarketingAutomation.Enrollment.InMemoryActivityEnrollmentCache>
```

- o In the `<Services>` section, after the `<MarketingAutomation.Enrollment.InMemoryActivityEnrollmentCache>` section, add the new `<MarketingAutomation.Enrollment.AutomationPlanFacetsUpdater>` section:

```
<!-- An updater that is used to update automation plan facets. -->
<MarketingAutomation.Enrollment.AutomationPlanFacetsUpdater>
<Type>Sitecore.Xdb.MarketingAutomation.Enrollment.AutomationPlanFacetsUpdater,
Sitecore.Xdb.MarketingAutomation</Type>
<As>Sitecore.Xdb.MarketingAutomation.Enrollment.IAutomationPlanFacetsUpdater,
Sitecore.Xdb.MarketingAutomation</As>
<LifeTime>Singleton</LifeTime>
</MarketingAutomation.Enrollment.AutomationPlanFacetsUpdater>
```

5. In the `sc.MarketingAutomation.ActivityTypes.xml` file, remove `<MarketingAutomation.Activity.ConditionalPlanEntry>` item that has the following ID:

```
<Id>{2EAF2623-BE5F-4DB7-8A29-BF50423E01BD}</Id>
```

6. In the `sc.MarketingAutomation.Processors.xml` file:

- o After the `</MarketingAutomation.Processing.LiveEventProcessor>` section, register the new `<MarketingAutomation.Processing.OperationResultCache>` service:

```
<!-- A cache used by processors to store intermediate results -->
<MarketingAutomation.Processing.OperationResultCache>
<Type>Sitecore.Xdb.MarketingAutomation.Cache.MemoryOperationResultCache,
Sitecore.Xdb.MarketingAutomation</Type>
<As>Sitecore.Xdb.MarketingAutomation.Core.Cache.IOperationResultCache,
Sitecore.Xdb.MarketingAutomation.Core</As>
<LifeTime>Singleton</LifeTime>
<Options>
  <!-- The duration entries will live in the cache for -->
  <EntryLifetime>0:00:30</EntryLifetime>
</Options>
```

```
</MarketingAutomation.Processing.OperationResultCache>
```

- In the `<XConnect.Service.Plugins.MarketingAutomation.XdbContactEventWatcher>` section, after the `<DefaultPriority>` option, add the following:

```
<!-- Enables or disables whether to retry failed operations when adding work items to the Automation Pool. Default: true -->  
<EnablePoolAddRetry>true</EnablePoolAddRetry>  
<!-- The maximum number of times to retry failed operations when adding work items to the Automation Pool. Default: 3 -->  
<PoolAddRetryLimit>3</PoolAddRetryLimit>
```

7. The following configuration files have been changed significantly:

- `sc.MarketingAutomation.ServiceDecorators.xml`
- `sc.MarketingAutomation.Workers.xml`

Save any customizations that you have made and replace the files with the new files from:

```
Sitecore 9.0.2 rev. 180604  
(OnPrem)_xp0xconnect.scwdp\Content\Website\App_data\config\sitecore\MarketingAutomation\
```

## Search Indexer Configuration

These configuration files are stored here:

```
xconnect\AppData\config\sitecore\SearchIndexer
```

1. In the `sc.Xdb.Collection.IndexerSettings.xml` file:

- In the `<IndexerSettings>` section, in the `<Options>` section, before the `<IndexPIISensitiveData>` node, add the new `<SplitRecordsThreshold>` section:

```
<!-- Indexer will split change set on chunks to improve memory consumption. Setting this option to 0, a negative value or removing the element completely, results in no splitting.-->  
<SplitRecordsThreshold>25000</SplitRecordsThreshold>
```

- After the `</IIndexer>` section, add the new `<IIndexerInitializer>` service:

```
<IIndexerInitializer>  
  <Type>Sitecore.Xdb.Collection.Indexing.DefaultIndexInitializer,  
  Sitecore.Xdb.Collection</Type>  
  <As>Sitecore.Xdb.Collection.Indexing.IIndexInitializer,  
  Sitecore.Xdb.Collection</As>  
  <LifeTime>Singleton</LifeTime>  
</IIndexerInitializer>
```

- In the `<IIndexRebuildFlow>` section, change the value of the `<BatchSize>` setting to:

```
<BatchSize>5000</BatchSize>
```

- Change the description of the connected `<ParallelizationDegree>` section to:

```
<!--Enable ParallelizationDegree setting to override default value which is (processorCount)-->
```

- After the `</IIndexRebuildFlow>` service, add the new `<IIndexRebuilderCountersDecorator>` service:

```
<IIndexRebuilderCountersDecorator>  
  <Type>Sitecore.XConnect.Configuration.ServiceDecorator`2[[Sitecore.Xdb.Collection.Indexing.IIndexRebuilder, Sitecore.Xdb.Collection], [Sitecore.Xdb.Collection.Indexing.IndexRebuilderCountersDecorator, Sitecore.Xdb.Collection]], Sitecore.XConnect.Configuration</Type>  
  <LifeTime>Singleton</LifeTime>  
</IIndexRebuilderCountersDecorator>
```

## 2. In the `sc.Xdb.Collection.IndexWriter.AzureSearch.xml.disabled` file:

- o In the `<Services>` section, after the `<service>` node, add the new `<IIndexingInitializer>` service:

```
<IIndexingInitializer>
<Type>Sitecore.Xdb.Collection.Search.Azure.IndexingInitializer,
Sitecore.Xdb.Collection.Search.Azure</Type>
<As>Sitecore.Xdb.Collection.Indexing.IIndexingInitializer,
Sitecore.Xdb.Collection</As>
<LifeTime>Singleton</LifeTime>
<Options>
<FailInitializationTimeoutMilliseconds>10000</FailInitializationTimeoutMillisecon
ds>
<RetryTimeoutMilliseconds>100</RetryTimeoutMilliseconds>
</Options>
</IIndexingInitializer>
```

- o In the `<Azure.AzureWriterSettings>` section, comment out the `<DataReplicationTimeoutMilliseconds>` section.

If you use Azure Search, you must enable this configuration file.

In Sitecore 9.0.2, this setting is disabled by default. If you use more than one replica, you must enable this setting

For more information about this setting, see the [developer documentation for Sitecore](#).

## Segmentation Configuration

These configuration files are stored here:

```
xconnect\App_data\config\sitecore\Segmentation
```

1. In the Sitecore 9.0.2 rev. 180604 (OnPrem) `_xp0xconnect.scwdp\Content\Website\App_data\config\sitecore\Segmentation` folder, copy the `sc.EXM.Segmentation.Predicates.xml` file and paste it into the `xconnect\App_data\config\sitecore\Segmentation` folder.

## xConnect Configuration

These configuration files are stored here:

```
xconnect\App_data\config\sitecore\XConnect
```

The `sc.XConnect.Client.xml.disabled` configuration file contains a sample that shows how the model can be extended.

- After the `<LifeTime>` section, in the `<XConnect.Client.Configuration>` section, add the new `<Options>` section:

```
<Options>
<!-- The custom xDB models to use with the xConnect client configuration -->
<Models>
  <!--
  <SampleModel>
  <TypeName>SampleAssembly.SampleModel, SampleAssembly</TypeName>
  <PropertyName>Model</PropertyName>
  </SampleModel>
  -->
</Models>
</Options>
```

## Messaging Configuration

- To add the new messaging configuration folder, copy the Sitecore 9.0.2 rev. 180604 (OnPrem) `_xp0xconnect.scwdp\Content\Website\App_data\config\sitecore\Messaging` folder and paste it into the `connect\App_data\config\sitecore\` folder.

## Diagnostic Counters Configuration

These configuration files are stored here:

```
xconnect\App_data\Diagnostics
```

Save any customizations that you have made.

1. Copy all the files from the Sitecore 9.0.2 rev. 180604 (OnPrem)\_xp0xconnect.scwdp\Content\Website\App\_data\Diagnostics folder and paste them into the xconnect\App\_data\Diagnostics folder.
2. Delete the following obsolete files from the xconnect\App\_data\Diagnostics folder:
  - o Sitecore.XConnect.Diagnostics.InstallCounters.pdb
  - o Sitecore.XConnect.Diagnostics.pdb
  - o Sitecore.XConnect.Diagnostics.PerformanceCounters.pdb

## XConnect Default Model Configuration

These configuration files are stored here:

```
xconnect\App_data\Models
```

1. In the folder xconnect\App\_data\Models folder, delete the obsolete Sitecore.XConnect.ContentTesting.Model, 1.0.json file.
2. In the Sitecore 9.0.2 rev. 180604 (OnPrem)\_xp0xconnect.scwdp\Content\Website\App\_data\Models folder, copy the new Sitecore.XConnect.ContentTesting.Model, 1.1.json content testing model file and paste it into the into xconnect\App\_data\Models folder.
3. In the Sitecore 9.0.2 rev. 180604 (OnPrem)\_xp0xconnect.scwdp\Content\Website\App\_data\Models folder, copy the following files:
  - o Sitecore.EmailCampaign.XConnectModel, 1.0.json
  - o Sitecore.XConnect.Collection.Model, 9.0.json
  - o Sitecore.XConnect.Collection.ContactMerge.Model, 1.0.jsonPaste them into the into xconnect\App\_data\Models folder.

## Collection Deployment Tool

1. In the xconnect\App\_data\collectiondeployment folder, delete all the files.
2. In the Sitecore 9.0.2 rev. 180604 (OnPrem)\_xp0xconnect.scwdp\Content\Website\App\_data\collectiondeployment folder, copy all the files and paste them into the into xconnect\App\_data\collectiondeployment folder.

## Bin Folder Update

1. In the Sitecore 9.0.2 rev. 180604 (OnPrem)\_xp0xconnect.scwdp\Content\Website\bin folder copy all the files and paste the files into the xconnect\bin folder of each xConnect instance that you want to upgrade.
2. Remove these obsolete files from the xconnect\bin folder:
  - o Sitecore.ContentTesting.Collection.Model.Plugins.dll
  - o Sitecore.ContentTesting.Model.dll
  - o Sitecore.Xdb.Collection.Data.SqlServer.dll.config
  - o Sitecore.Xdb.Common.Web.dll.config

- o Sitecore.Xdb.ReferenceData.Service.dll.config
- o Sitecore.Xdb.ReferenceData.SqlServer.dll.config
- o Sitecore.Xdb.ReferenceData.Web.dll.config
- o Sitecore.Xdb.ReferenceData.Web.XConnect.dll.config
- o Sitecore.Xdb.Sql.Common.dll.config

## 8.2.1 Upgrade of the Sitecore XConnect Search Indexer service

If you are using the Solr search provider:

- In the `xconnect\App_data\jobs\continuous\IndexWorker\App_config\ConnectionStrings.config` file, update the Sitecore XConnect Search Indexer "solrCore" connection string and ensure that it points to the xdb core that is attached to Solr 6.6.2.

### Sitecore XConnect Search Indexer Configuration

1. Stop the *Sitecore XConnect Search Indexer* Windows service.
2. In the Sitecore 9.0.2 rev. 180604 (OnPrem)\_xp0xconnect.scwdp\Content\Website\App\_data\jobs\continuous\IndexWorker folder, copy all the files and paste them into the `xconnect\App_data\jobs\continuous\IndexWorker` folder.
3. In the `xconnect\App_data\jobs\continuous\IndexWorker\App_data\Models` folder, delete the obsolete `Sitecore.XConnect.ContentTesting.Model, 1.0.json` file.
4. In the Sitecore 9.0.2 rev. 180604 (OnPrem)\_xp0xconnect.scwdp\Content\Website\App\_data\jobs\continuous\IndexWorker\App\_data\Models folder, copy the following files:
  - o `Sitecore.XConnect.ContentTesting.Model, 1.1.json`
  - o `Sitecore.XConnect.Collection.Model, 9.0.json`
  - o `Sitecore.XConnect.Collection.ContactMerge.Model, 1.0.json`
  - o `Sitecore.EmailCampaign.XConnectModel, 1.0.json`Paste them into the `xconnect\App_data\jobs\continuous\IndexWorker\App_data\Models` folder.
5. In the `xconnect\App_data\jobs\continuous\IndexWorker` folder, delete the obsolete `XConnectSearchIndexer.pdb` file.

### Collection Service Configuration

The configuration files are stored here:

```
xconnect\App_data\jobs\continuous\IndexWorker\App_data\Config\Sitecore\Collection
```

1. In the `sc.Xdb.Collection.RepositorySettings.xml` file:
  - o In the `<IRepository>` section, change the `<Type>` value to:

```
<Type>Sitecore.Xdb.Collection.Repository, Sitecore.Xdb.Collection</Type>
```
  - o In the `<Services>` section, after the `<IRepository>` section, add the new `<IRepositoryCountersDecorator>` section:

```
<IRepositoryCountersDecorator>
<Type>Sitecore.XConnect.Configuration.ServiceDecorator`2[[Sitecore.Xdb.Collection.IRepository, Sitecore.Xdb.Collection],
[Sitecore.Xdb.Collection.RepositoryCountersDecorator, Sitecore.Xdb.Collection]],
Sitecore.XConnect.Configuration</Type>
<LifeTime>Singleton</LifeTime>
```

```
<Options>
<!-- Level number (0..255) that is used to sort decorators of the same service. -
->
  <LevelNumber>10</LevelNumber>
</Options>
</IRepositoryCountersDecorator>
```

- o In the `<Services>` section, after the `<IRepositoryCountersDecorator>` section, add the new `<IDataProviderCountersDecorator>` section:

```
<IDataProviderCountersDecorator>
<Type>Sitecore.XConnect.Configuration.ServiceDecorator`2[[Sitecore.Xdb.Collection.I
DataProvider, Sitecore.Xdb.Collection],
[Sitecore.Xdb.Collection.DataProviderCountersDecorator, Sitecore.Xdb.Collection]],
Sitecore.XConnect.Configuration</Type>
  <LifeTime>Singleton</LifeTime>
  <Options>
  <!-- Level number (0..255) that is used to sort decorators of the same service. -
  ->
    <LevelNumber>10</LevelNumber>
  </Options>
</IDataProviderCountersDecorator>
```

## Search Service Configuration

The configuration files are stored here:

```
xconnect\App_data\jobs\continuous\IndexWorker\App_data\Config\Sitecore\CollectionSearch
```

1. In the `sc.Xdb.Collection.IndexReader.AzureSearch.xml.disabled` file, in the `<Services>` section, after the `<IAzureWebClientFactory>` node, add the new `<IAzureSecondaryWebClientFactory>` service:

```
<IAzureSecondaryWebClientFactory>
<Type>Sitecore.Xdb.Collection.Search.Azure.WebClient.SecondaryCoreAzureWebClientFactor
y, Sitecore.Xdb.Collection.Search.Azure</Type>
<As>Sitecore.Xdb.Collection.Search.Azure.WebClient.ISecondaryWebClientFactory,
Sitecore.Xdb.Collection.Search.Azure</As>
  <LifeTime>Singleton</LifeTime>
  <Options>
    <ConnectionStringName>
      <!-- Set connection string name if you need to use different connection
string or different from default core name.
collection.search.secondary-->
    </ConnectionStringName>
  </Options>
</IAzureSecondaryWebClientFactory>
```

## Core Services Configuration

The configuration files are stored here:

```
xconnect\App_data\jobs\continuous\IndexWorker\App_data\Config\Sitecore\CoreServices
```

1. In the `sc.Xdb.Sql.Common.xml` file:
  - o In the `<RetryManager>` section, remove the `<Options>` section.
  - o In the `<Services>` section, before the `<RetryManager>` section, add the new `<RetryManagerSettings>` section:

```
<RetryManagerSettings>
  <Type>Sitecore.Xdb.Sql.Common.ExponentialBackoffRetryManagerSettings,
Sitecore.Xdb.Sql.Common</Type>
  <As>Sitecore.Xdb.Sql.Common.IRetryManagerSettings, Sitecore.Xdb.Sql.Common</As>
  <LifeTime>Transient</LifeTime>
  <Options>
    <RetryCount>10</RetryCount>
    <MinBackoff>00:00:01</MinBackoff>
    <MaxBackoff>00:00:30</MaxBackoff>
    <DeltaBackoff>00:00:10</DeltaBackoff>
  </Options>
</RetryManagerSettings>
```

## Search Indexer Configuration

The configuration files are stored here:

```
xconnect\AppData\jobs\continuous\IndexWorker\AppData\Config\Sitecore\SearchIndexer
```

### 1. In the `sc.Xdb.Collection.IndexerSettings.xml` file:

- o In the `<IndexerSettings>` section, in the `<Options>` section, before `<IndexPIISensitiveData>` node, add the new `<SplitRecordsThreshold>` section:

```
<!-- Indexer will split change set on chunks to improve memory consumption.
Setting this option to 0, a negative value or removing the element completely,
results in no splitting.-->
<SplitRecordsThreshold>25000</SplitRecordsThreshold>
```

- o After the `</IIndexer>` section, add the new `<IIndexerInitializer>` service:

```
<IIndexerInitializer>
  <Type>Sitecore.Xdb.Collection.Indexing.DefaultIndexInitializer,
  Sitecore.Xdb.Collection</Type>
  <As>Sitecore.Xdb.Collection.Indexing.IIndexInitializer,
  Sitecore.Xdb.Collection</As>
  <LifeTime>Singleton</LifeTime>
</IIndexerInitializer>
```

- o In the `<IIndexRebuildFlow>` section, change the value of the `<BatchSize>` setting to:

```
<BatchSize>5000</BatchSize>
```

- o Change the description for the connected `<ParallelizationDegree>` section to:

```
<!--Enable ParallelizationDegree setting to override default value which is
(processorCount)-->
```

- o After the `</IIndexRebuildFlow>` section, add the new `<IIndexRebuilderCountersDecorator>` service:

```
<IIndexRebuilderCountersDecorator>
<Type>Sitecore.XConnect.Configuration.ServiceDecorator`2[[Sitecore.Xdb.Collection.I
ndexing.IIndexRebuilder, Sitecore.Xdb.Collection],
[Sitecore.Xdb.Collection.Indexing.IndexRebuilderCountersDecorator,
Sitecore.Xdb.Collection]], Sitecore.XConnect.Configuration</Type>
  <LifeTime>Singleton</LifeTime>
</IIndexRebuilderCountersDecorator>
```

### 2. In the `sc.Xdb.Collection.IndexWriter.AzureSearch.xml.disabled` file:

- o In the `<Services>` section, after the `<Azure.AzureWriterSettings>` node, add the new `<IIndexingInitializer>` service:

```
<IIndexingInitializer>
<Type>Sitecore.Xdb.Collection.Search.Azure.IndexingInitializer,
Sitecore.Xdb.Collection.Search.Azure</Type>
<As>Sitecore.Xdb.Collection.Indexing.IIndexInitializer,
Sitecore.Xdb.Collection</As>
<LifeTime>Singleton</LifeTime>
<Options>
<FailInitializationTimeoutMilliseconds>100000</FailInitializationTimeoutMillisecond
s>
<RetryTimeoutMilliseconds>100</RetryTimeoutMilliseconds>
</Options>
</IIndexingInitializer>
```

- o In the `<Azure.AzureWriterSettings>` section, comment out the `<DataReplicationTimeoutMilliseconds>` node.

If you use Azure Search, you must enable this configuration file.

In Sitecore 9.0.2, this setting is disabled by default. If you use more than one replica, you must enable this setting

For more information about this setting, see the [developer documentation for Sitecore](#).

## 8.2.2 Upgrade the Sitecore Marketing Automation Engine service

1. Stop the *Sitecore Marketing Automation Engine* Windows service.
2. In the Sitecore 9.0.2 rev. 180604 (OnPrem) `_xp0xconnect.scwdp\Content\Website\App_data\jobs\continuous\AutomationEngine` folder, copy all the files – not the folders – and paste the files into the `xconnect\App_data\jobs\continuous\AutomationEngine` folder.
3. In the root folder of the service, delete the following obsolete files:
  - o `maengine.pdb`
  - o `Sitecore.ContentTesting.Collection.Model.Plugins.dll`
  - o `Sitecore.ContentTesting.Model.dll`
  - o `Sitecore.Xdb.Common.Web.dll.config`
  - o `Sitecore.Xdb.MarketingAutomation.pdb`
  - o `Sitecore.Xdb.ReferenceData.Service.dll.config`
  - o `Sitecore.Xdb.MarketingAutomation.SqlServer.Testing.Common.dll`
  - o `Sitecore.Xdb.ReferenceData.SqlServer.dll.config`
  - o `Sitecore.Xdb.Sql.Common.dll.config`
4. In the `xconnect\App_data\jobs\continuous\AutomationEngine\App_Config` folder, in the `ConnectionStrings.config` file, add the following connection string:

```
<add name="messaging" connectionString="user id=user;password=password;data source=source;Initial Catalog=dbname" />
```

### Marketing Automation Services Configuration

The configuration files are stored here:

```
xconnect\App_data\jobs\continuous\AutomationEngine\App_Data\Config\sitecore\MarketingAutomation
```

1. In the Sitecore 9.0.2 rev. 180604 (OnPrem) `_xp0xconnect.scwdp\Content\Website\App_data\jobs\continuous\AutomationEngine\App_Data\Config\sitecore\MarketingAutomation` folder, copy the following files:
  - o `sc.MarketingAutomation.PoolFilter.xml`
  - o `sc.EXM.ActivityTypes.xml`

Paste them into the

`xconnect\App_data\jobs\continuous\AutomationEngine\App_Data\Config\sitecore\MarketingAutomation` folder.

2. In the `sc.MarketingAutomation.ActivityResultHandlers.xml` file, in the `<MarketingAutomation.Factory.ActivityResultHandlerFactory>` section, change:

```
<LifeTime>Singleton</LifeTime>
```

to:

```
<LifeTime>Scoped</LifeTime>
```

3. In the `sc.MarketingAutomation.Locators.xml` file, after the `<MarketingAutomation.Locator.MarketingsOperationsPlanLocator>` into `<Services>` section, add the new `<MarketingAutomation.Locator.MemoryPlanDefinitionCache>` section:

```
<!-- Cache that stores plan definitions in memory. -->  
<MarketingAutomation.Locator.MemoryPlanDefinitionCache>
```

```
<Type>Sitecore.Xdb.MarketingAutomation.Locator.MemoryPlanDefinitionCache,
Sitecore.Xdb.MarketingAutomation</Type>
<As>Sitecore.Xdb.MarketingAutomation.Locator.IPlanDefinitionCache,
Sitecore.Xdb.MarketingAutomation</As>
<LifeTime>Singleton</LifeTime>
<Options>
  <!-- The lifetime of entries in the cache. -->
  <EntryLifetime>00:00:20</EntryLifetime>
</Options>
</MarketingAutomation.Locator.MemoryPlanDefinitionCache>
```

4. In the `sc.MarketingAutomation.PlanProcessing.xml` file:

- o In the `<Services>` section, after the `</MarketingAutomation.Processing.Results.ResultPrecedenceResultAggregator>` section, add the new `<MarketingAutomation.Enrollment.InMemoryActivityEnrollmentCache>` section:

```
<!-- A cache that stores activity enrollments in memory during a work item
processing. -->
<MarketingAutomation.Enrollment.InMemoryActivityEnrollmentCache>

  <Type>Sitecore.Xdb.MarketingAutomation.Enrollment.InMemoryActivityEnrollmentCache
, Sitecore.Xdb.MarketingAutomation</Type>
  <As>Sitecore.Xdb.MarketingAutomation.Enrollment.IActivityEnrollmentCache,
Sitecore.Xdb.MarketingAutomation</As>
  <LifeTime>Scoped</LifeTime>
</MarketingAutomation.Enrollment.InMemoryActivityEnrollmentCache>
```

- o In the `<Services>` section, after the `<MarketingAutomation.Enrollment.InMemoryActivityEnrollmentCache>` section, add the new `<MarketingAutomation.Enrollment.AutomationPlanFacetsUpdater>` section:

```
<!-- An updater that is used to update automation plan facets. -->
<MarketingAutomation.Enrollment.AutomationPlanFacetsUpdater>
  <Type>Sitecore.Xdb.MarketingAutomation.Enrollment.AutomationPlanFacetsUpdater,
Sitecore.Xdb.MarketingAutomation</Type>
  <As>Sitecore.Xdb.MarketingAutomation.Enrollment.IAutomationPlanFacetsUpdater,
Sitecore.Xdb.MarketingAutomation</As>
  <LifeTime>Singleton</LifeTime>
</MarketingAutomation.Enrollment.AutomationPlanFacetsUpdater>
```

5. In the `sc.MarketingAutomation.ActivityTypes.xml` file, in the `<MarketingAutomation.Activity.ConditionalPlanEntry>` section, remove the node with the following ID:

```
<Id>{2EAF2623-BE5F-4DB7-8A29-BF50423E01BD}</Id>
```

6. In the `sc.MarketingAutomation.Processors.xml` file, after the `</MarketingAutomation.Processing.LiveEventProcessor>` section, add the new `<MarketingAutomation.Processing.OperationResultCache>` service:

```
<!-- A cache used by processors to store intermediate results -->
<MarketingAutomation.Processing.OperationResultCache>
<Type>Sitecore.Xdb.MarketingAutomation.Cache.MemoryOperationResultCache,
Sitecore.Xdb.MarketingAutomation</Type>
<As>Sitecore.Xdb.MarketingAutomation.Core.Cache.IOperationResultCache,
Sitecore.Xdb.MarketingAutomation.Core</As>
<LifeTime>Singleton</LifeTime>
<Options>
  <!-- The duration entries will live in the cache for -->
  <EntryLifetime>0:00:30</EntryLifetime>
</Options>
</MarketingAutomation.Processing.OperationResultCache>
```

7. The following configuration files have changed significantly:

- o `sc.MarketingAutomation.ServiceDecorators.xml`
- o `sc.MarketingAutomation.Workers.xml`

# Sitecore Experience Platform Upgrade Guide

Save any customizations that you have made and replace the files with the new files from:

```
Sitecore 9.0.2 rev. 180604  
(OnPrem)_xp0xconnect.scwdp\Content\Website\App_data\jobs\continuous\AutomationEngine\App_Data\Config\sitecore\MarketingAutomation
```

## Core Services Configuration

The configuration files are stored here:

```
xconnect\AppData\jobs\continuous\AutomationEngine\AppData\Config\sitecore\CoreServices
```

1. In the `sc.Xdb.Sql.Common.xml` file:
  - o In the `<RetryManager>` section, delete the `<Options>` node.
  - o In the `<Services>` section, before the `<RetryManager>` section, add the new `<RetryManagerSettings>` section:

```
<RetryManagerSettings>  
  <Type>Sitecore.Xdb.Sql.Common.ExponentialBackoffRetryManagerSettings,  
  Sitecore.Xdb.Sql.Common</Type>  
  <As>Sitecore.Xdb.Sql.Common.IRetryManagerSettings, Sitecore.Xdb.Sql.Common</As>  
  <LifeTime>Transient</LifeTime>  
  <Options>  
    <RetryCount>10</RetryCount>  
    <MinBackoff>00:00:01</MinBackoff>  
    <MaxBackoff>00:00:30</MaxBackoff>  
    <DeltaBackoff>00:00:10</DeltaBackoff>  
  </Options>  
</RetryManagerSettings>
```

## Segmentation Configuration

The configuration files are stored here:

```
xconnect\AppData\jobs\continuous\AutomationEngine\AppData\Config\sitecore\Segmentation
```

2. In the Sitecore 9.0.2 rev. 180604  
(OnPrem)\_xp0xconnect.scwdp\Content\Website\App\_data\jobs\continuous\AutomationEngine\App\_Data\Config\sitecore\Segmentation folder, copy the `sc.EXM.Segmentation.Predicates.xml` file and paste it into the `xconnect\AppData\jobs\continuous\AutomationEngine\AppData\Config\sitecore\Segmentation` folder.

## Messaging Configuration

A new messaging configuration folder has been added.

- Copy the Sitecore 9.0.2 rev. 180604  
(OnPrem)\_xp0xconnect.scwdp\Content\Website\App\_data\jobs\continuous\AutomationEngine\App\_Data\Config\sitecore\Messaging folder and paste it into the `xconnect\AppData\jobs\continuous\AutomationEngine\AppData\Config\sitecore` folder.

## XConnect Configuration

The configuration files are stored here:

```
xconnect\AppData\jobs\continuous\AutomationEngine\AppData\Config\sitecore\XConnect
```

The `sc.XConnect.Client.xml` now contains an example of how the model can be extended.

- In the `<XConnect.Client.Configuration>` section, after the `<LifeTime>` node, add the new `<Options>` section:

```
<Options>  
<!-- The custom xDB models to use with the xConnect client configuration -->  
<Models>  
  <!--  
  <SampleModel>  
  <TypeName>SampleAssembly.SampleModel, SampleAssembly</TypeName>
```

```
<PropertyName>Model</PropertyName>  
</SampleModel>  
-->  
</Models>  
</Options>
```

## 8.2.3 Resuming Services

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

## Chapter 9

# Upgrade xConnect from Sitecore XP 9.0.1 to Sitecore XP 9.0 Update 2

This chapter describes how to upgrade xConnect when you upgrade from Sitecore XP 9.0.1 to Sitecore XP 9.0 Update 2.

If you are upgrading from Sitecore XP 9.0.1 with xDB enabled, you must perform all the tasks described in this chapter.

This chapter contains the following sections:

- Prepare for the Upgrade
- Configuration Changes

## 9.1 Prepare for the Upgrade

### 9.1.1 Prerequisites for Upgrading xConnect

Before you can start the upgrade process, you must ensure that you have all the necessary packages and scripts.

If you are going to move from SQL Server 2016 to MongoDB as part of the upgrade, the installation guide for Sitecore XP 9.0 Update 2 describes how to configure the MongoDB provider for xConnect.

To prepare for the upgrade:

1. [Download the packages for XP Single – Sitecore 9.0.2 rev. 180604 \(WDP XP0 packages\)](#) .zip – and unpack the zip file.

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

- o Sitecore 9.0.2 rev. 180604 (OnPrem)\_xp0xconnect.scwdp.zip
2. Unpack the Sitecore 9.0.2 rev. 180604 (OnPrem)\_xp0xconnect.scwdp.zip file so that the files it contains are ready for use.
  3. Unpack the Database Upgrade Script.zip file.

It contains several database upgrade scripts including:

- o SXP90\_db\_marketingautomation\_BeforeInstall.sql
  - o SXP90\_db\_processing\_pools\_BeforeInstall.sql
  - o SXP90\_db\_collection\_BeforeInstall.sql
4. If you are using Solr provider, ensure that you have updated Solr to version 6.6.2.
  5. If you have any functionality that accesses the contact/interaction data extraction process from outside of xConnect, you must stop it. You can restart it when the upgrade is completed. An example of this kind of functionality is rebuilding the *Reporting* database.

#### Important

You cannot just pause these tasks before the upgrade and resume them afterwards because the extraction cursors have changed in Sitecore XP 9.0.2.

### 9.1.2 Upgrade the Databases

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

To upgrade the databases:

1. Execute the `SXP90_db_marketingautomation_BeforeInstall.sql` script for the *Xdb Marketing Automation* database.
2. Execute the `SXP90_db_processing_pools_BeforeInstall.sql` script for the *Xdb Processing Pools* database

## 9.2 Configuration Changes

You must make the following configuration changes.

### Collection Service Configuration

The configuration files are stored here:

```
xconnect\AppData\config\sitecore\Collection
```

1. In the `sc.Xdb.Collection.RepositorySettings.xml` file:

- o In the `<IRepository>` section, change the `<Type>` value to :

```
<Type>Sitecore.Xdb.Collection.Repository, Sitecore.Xdb.Collection</Type>
```

- o In the `<IRepository>` section, add the `<IRepositoryCountersDecorator>` section:

```
<IRepositoryCountersDecorator>
<Type>Sitecore.XConnect.Configuration.ServiceDecorator`2[[Sitecore.Xdb.Collection.IRepository, Sitecore.Xdb.Collection],
[Sitecore.Xdb.Collection.RepositoryCountersDecorator, Sitecore.Xdb.Collection]],
Sitecore.XConnect.Configuration</Type>
  <LifeTime>Singleton</LifeTime>
  <Options>
  <!-- Level number (0..255) that is used to sort decorators of the same service. -->
    <LevelNumber>10</LevelNumber>
  </Options>
</IRepositoryCountersDecorator>
```

- o In the `<IRepositoryCountersDecorator>` section, add the `<IDataProviderCountersDecorator>` section:

```
<IDataProviderCountersDecorator>
<Type>Sitecore.XConnect.Configuration.ServiceDecorator`2[[Sitecore.Xdb.Collection.IDataProvider, Sitecore.Xdb.Collection],
[Sitecore.Xdb.Collection.DataProviderCountersDecorator, Sitecore.Xdb.Collection]],
Sitecore.XConnect.Configuration</Type>
  <LifeTime>Singleton</LifeTime>
  <Options>
  <!-- Level number (0..255) that is used to sort decorators of the same service. -->
    <LevelNumber>10</LevelNumber>
  </Options>
</IDataProviderCountersDecorator>
```

2. In the Sitecore 9.0.2 rev. 180604 (OnPrem) `_xp0xconnect.scwdp\Content\Website\AppData\config\sitecore\Collection` folder that you unpacked earlier, copy the `sc.Xdb.Collection.Data.MongoDb.xml.disabled` file and paste it into the `xconnect\AppData\config\sitecore\Collection` folder.

### Search Service Configuration

The configuration files are stored here:

```
xconnect\AppData\Config\Sitecore\CollectionSearch
```

1. In the `sc.Xdb.Collection.IndexReader.AzureSearch.xml.disabled` file, in the `<Services>` section, after the `<IAzureWebClientFactory>` node, add the new `<IAzureSecondaryWebClientFactory>` service:

```
<IAzureSecondaryWebClientFactory>
<Type>Sitecore.Xdb.Collection.Search.Azure.WebClient.SecondaryCoreAzureWebClientFactory, Sitecore.Xdb.Collection.Search.Azure</Type>
<As>Sitecore.Xdb.Collection.Search.Azure.WebClient.ISecondaryWebClientFactory, Sitecore.Xdb.Collection.Search.Azure</As>
  <LifeTime>Singleton</LifeTime>
  <Options>
    <ConnectionStringName>
      <!-- Set connection string name if you need to use different connection string or different from default core name.
      collection.search.secondary-->
    </ConnectionStringName>
  </Options>
```

```
</IAzureSecondaryWebClientFactory>
```

## Marketing Automation Services Configuration

These configuration files are stored here:

```
xconnect\App_data\config\sitecore\MarketingAutomation
```

1. In the `sc.MarketingAutomation.ActivityTypes.xml` file, remove the `<MarketingAutomation.Activity.ConditionalPlanEntry>` item that has the following ID:

```
<Id>{2EAF2623-BE5F-4DB7-8A29-BF50423E01BD}</Id>
```

2. In the `sc.MarketingAutomation.Processors.xml` file, after the `</MarketingAutomation.Processing.LiveEventProcessor>` section, register the new `<MarketingAutomation.Processing.OperationResultCache>` service:

```
<!-- A cache used by processors to store intermediate results -->
<MarketingAutomation.Processing.OperationResultCache>
  <Type>Sitecore.Xdb.MarketingAutomation.Cache.MemoryOperationResultCache,
  Sitecore.Xdb.MarketingAutomation</Type>
  <As>Sitecore.Xdb.MarketingAutomation.Core.Cache.IOperationResultCache,
  Sitecore.Xdb.MarketingAutomation.Core</As>
  <LifeTime>Singleton</LifeTime>
  <Options>
    <!-- The duration entries will live in the cache for -->
    <EntryLifetime>0:00:30</EntryLifetime>
  </Options>
</MarketingAutomation.Processing.OperationResultCache>
```

3. In the `sc.XConnect.Service.Plugins.MarketingAutomation.xml` file, in the `<XConnect.Service.Plugins.MarketingAutomation.XdbContactEventWatcher>` section, after the `<DefaultPriority>` option, add the following:

```
<!-- Enables or disables whether to retry failed operations when adding work items to
the Automation Pool. Default: true -->
<EnablePoolAddRetry>true</EnablePoolAddRetry>
<!-- The maximum number of times to retry failed operations when adding work items to
the Automation Pool. Default: 3 -->
<PoolAddRetryLimit>3</PoolAddRetryLimit>
```

## Search Indexer Configuration

These configuration files are stored here:

```
xconnect\App_data\config\sitecore\SearchIndexer
```

1. In the `sc.Xdb.Collection.IndexerSettings.xml` file:
  - o After the `</IIndexer>` section, add the new `<IIndexerInitializer>` service:

```
<IIndexerInitializer>
  <Type>Sitecore.Xdb.Collection.Indexing.DefaultIndexInitializer,
  Sitecore.Xdb.Collection</Type>
  <As>Sitecore.Xdb.Collection.Indexing.IIndexInitializer,
  Sitecore.Xdb.Collection</As>
  <LifeTime>Singleton</LifeTime>
</IIndexerInitializer>
```

- o In the `<IIndexRebuildFlow>` section, change the value of the `<BatchSize>` setting to:

```
<BatchSize>5000</BatchSize>
```

- o Change the description of the connected `<ParallelizationDegree>` section to:

```
<!--Enable ParallelizationDegree setting to override default value which is
(processorCount)-->
```

- o After the `</IIndexRebuildFlow>` service, add the new `<IIndexRebuilderCountersDecorator>` service:

```
<IIndexRebuilderCountersDecorator>
  <Type>Sitecore.XConnect.Configuration.ServiceDecorator`2[[Sitecore.Xdb.Collection
  .Indexing.IIndexRebuilder, Sitecore.Xdb.Collection],
```

```
[Sitecore.Xdb.Collection.Indexing.IndexRebuilderCountersDecorator,  
Sitecore.Xdb.Collection]], Sitecore.XConnect.Configuration</Type>  
<LifeTime>Singleton</LifeTime>  
</IIndexRebuilderCountersDecorator>
```

2. In the `sc.Xdb.Collection.IndexWriter.AzureSearch.xml.disabled` file, in the `<services>` section, after the `<Azure.AzureWriterSettings>` node, add the new `<IIndexingInitializer>` service:

```
<IIndexingInitializer>  
<Type>Sitecore.Xdb.Collection.Search.Azure.IndexingInitializer,  
Sitecore.Xdb.Collection.Search.Azure</Type>  
<As>Sitecore.Xdb.Collection.Indexing.IIndexInitializer, Sitecore.Xdb.Collection</As>  
<LifeTime>Singleton</LifeTime>  
<Options>  
<FailInitializationTimeoutMilliseconds>100000</FailInitializationTimeoutMilliseconds>  
<RetryTimeoutMilliseconds>100</RetryTimeoutMilliseconds>  
</Options>  
</IIndexingInitializer>
```

## Messaging Configuration

1. In the Sitecore 9.0.2 rev. 180604 (OnPrem) `_xp0xconnect.scwdp\Content\Website\App_data\config\sitecore\Messaging` folder that you unpacked earlier, copy the following files:
  - o `sc.XConnect.TransientFaultHandling.xml`
  - o `sc.XConnect.Messaging.MyFeature.Sql.xml.example`Paste them into the `xconnect\App_data\config\sitecore\Messaging` folder.

## Diagnostic Counters Configuration

These configuration files are stored here:

```
xconnect\App_data\Diagnostics
```

Save any customizations that you have made.

1. Copy all the files from the Sitecore 9.0.2 rev. 180604 (OnPrem) `_xp0xconnect.scwdp\Content\Website\App_data\Diagnostics` folder and paste them into the `xconnect\App_data\Diagnostics` folder.

## XConnect Default Model Configuration

These configuration files are stored here:

```
xconnect\App_data\Models
```

1. In the Sitecore 9.0.2 rev. 180604 (OnPrem) `_xp0xconnect.scwdp\Content\Website\App_data\Models` folder, copy the following files:
  - o `Sitecore.XConnect.Collection.ContactMerge.Model, 1.0.json`
  - o `Sitecore.XConnect.ContentTesting.Model, 1.1.json`Paste them into the `xconnect\App_data\Models` folder.

## Bin Folder Update

- In the Sitecore 9.0.2 rev. 180604 (OnPrem) `_xp0xconnect.scwdp\Content\Website\bin` folder, copy all the files and paste them into the `xconnect\bin` folder of each XConnect instance that you want to upgrade.

## 9.2.1 Upgrade of the Sitecore XConnect Search Indexer service

### Sitecore XConnect Search Indexer Configuration

1. In the Sitecore 9.0.2 rev. 180604 (OnPrem)\_xp0xconnect.scwdp\Content\Website\App\_data\jobs\continuous\IndexWorker folder, copy all the files except the App\_config and App\_data folders and paste them into the into the xconnect\App\_data\jobs\continuous\IndexWorker folder.
2. In the Sitecore 9.0.2 rev. 180604 (OnPrem)\_xp0xconnect.scwdp\Content\Website\App\_data\jobs\continuous\IndexWorker\App\_data\Models folder, copy the following new files:
  - o Sitecore.XConnect.Collection.ContactMerge.Model, 1.0.json
  - o Sitecore.XConnect.ContentTesting.Model, 1.1.json

Paste them into the

xconnect\App\_data\jobs\continuous\IndexWorker\App\_data\Models folder.

3. Copy all the files from the Sitecore 9.0.2 rev. 180604 (OnPrem)\_xp0xconnect.scwdp\Content\Website\App\_data\jobs\continuous\IndexWorker\App\_data\Diagnostics folder and paste them into the xconnect\App\_data\jobs\continuous\IndexWorker\App\_data\Diagnostics folder.

### Collection Service Configuration

The configuration files are stored here:

```
xconnect\App_data\jobs\continuous\IndexWorker\App_data\Config\Sitecore\Collection
```

1. In the sc.Xdb.Collection.RepositorySettings.xml file,

- o In the <IRepository> section, change the <Type> value to:

```
<Type>Sitecore.Xdb.Collection.Repository, Sitecore.Xdb.Collection</Type>
```

- o In the <Services> section, after the <IRepository> section, add the new <IRepositoryCountersDecorator> section:

```
<IRepositoryCountersDecorator>  
<Type>Sitecore.XConnect.Configuration.ServiceDecorator`2[[Sitecore.Xdb.Collection.IRepository, Sitecore.Xdb.Collection],  
[Sitecore.Xdb.Collection.RepositoryCountersDecorator, Sitecore.Xdb.Collection]],  
Sitecore.XConnect.Configuration</Type>  
<LifeTime>Singleton</LifeTime>  
<Options>  
<!-- Level number (0..255) that is used to sort decorators of the same service. -->  
<LevelNumber>10</LevelNumber>  
</Options>  
</IRepositoryCountersDecorator>
```

- o In the <Services> section, after the <IRepositoryCountersDecorator> section, add the new <IDataProviderCountersDecorator> section:

```
<IDataProviderCountersDecorator>  
<Type>Sitecore.XConnect.Configuration.ServiceDecorator`2[[Sitecore.Xdb.Collection.IDataProvider, Sitecore.Xdb.Collection],  
[Sitecore.Xdb.Collection.DataProviderCountersDecorator, Sitecore.Xdb.Collection]],  
Sitecore.XConnect.Configuration</Type>  
<LifeTime>Singleton</LifeTime>  
<Options>  
<!-- Level number (0..255) that is used to sort decorators of the same service. -->  
<LevelNumber>10</LevelNumber>  
</Options>  
</IDataProviderCountersDecorator>
```

2. In the Sitecore 9.0.2 rev. 180604 (OnPrem)\_xp0xconnect.scwdp\Content\Website\App\_data\jobs\continuous\I

ndexWorker\AppData\Config\Sitecore\Collection folder that you unpacked earlier, copy the `sc.Xdb.Collection.Data.MongoDb.xml.disabled` file and paste it into the `xconnect\AppData\jobs\continuous\IndexWorker\AppData\Config\Sitecore\Collection` folder.

## Search Service Configuration

The configuration files are stored here:

```
xconnect\AppData\jobs\continuous\IndexWorker\AppData\Config\Sitecore\CollectionSearch
```

1. In the `sc.Xdb.Collection.IndexReader.AzureSearch.xml.disabled` file, in the `<Services>` section, after the `<IAzureWebClientFactory>` node, add the new `<IAzureSecondaryWebClientFactory>` service:

```
<IAzureSecondaryWebClientFactory>
<Type>Sitecore.Xdb.Collection.Search.Azure.WebClient.SecondaryCoreAzureWebClientFactory, Sitecore.Xdb.Collection.Search.Azure</Type>
<As>Sitecore.Xdb.Collection.Search.Azure.WebClient.ISecondaryWebClientFactory, Sitecore.Xdb.Collection.Search.Azure</As>
  <LifeTime>Singleton</LifeTime>
  <Options>
    <ConnectionStringName>
      <!-- Set connection string name if you need to use different connection string or different from default core name.
      collection.search.secondary-->
    </ConnectionStringName>
  </Options>
</IAzureSecondaryWebClientFactory>
```

## Search Indexer Configuration

The configuration files are stored here:

```
xconnect\AppData\jobs\continuous\IndexWorker\AppData\Config\Sitecore\SearchIndexer
```

1. In the `sc.Xdb.Collection.IndexerSettings.xml` file:
  - o After the `<IIndexer>` section, register the new `<IIndexerInitializer>` service:

```
<IIndexerInitializer>
  <Type>Sitecore.Xdb.Collection.Indexing.DefaultIndexInitializer, Sitecore.Xdb.Collection</Type>
  <As>Sitecore.Xdb.Collection.Indexing.IIndexInitializer, Sitecore.Xdb.Collection</As>
  <LifeTime>Singleton</LifeTime>
</IIndexerInitializer>
```

- o In the `<IIndexRebuildFlow>` section, change the value of the `<BatchSize>` setting to:

```
<BatchSize>5000</BatchSize>
```

- o Change description for the `<ParallelizationDegree>` section to:

```
<!--Enable ParallelizationDegree setting to override default value which is (processorCount)-->
```

- o After the `</IIndexRebuildFlow>` section, register the new `<IIndexRebuilderCountersDecorator>` service:

```
<IIndexRebuilderCountersDecorator>
  <Type>Sitecore.XConnect.Configuration.ServiceDecorator`2[[Sitecore.Xdb.Collection.Indexing.IIndexRebuilder, Sitecore.Xdb.Collection], [Sitecore.Xdb.Collection.Indexing.IndexRebuilderCountersDecorator, Sitecore.Xdb.Collection]], Sitecore.XConnect.Configuration</Type>
  <LifeTime>Singleton</LifeTime>
</IIndexRebuilderCountersDecorator>
```

2. In the `sc.Xdb.Collection.IndexWriter.AzureSearch.xml.disabled` file, in the `<Services>` section, after the `<Azure.AzureWriterSettings>` node, register the new `<IIndexingInitializer>` service:

```
<IIndexingInitializer>
<Type>Sitecore.Xdb.Collection.Search.Azure.IndexingInitializer, Sitecore.Xdb.Collection.Search.Azure</Type>
```

```
<As>Sitecore.Xdb.Collection.Indexing.IIndexInitializer, Sitecore.Xdb.Collection</As>
<LifeTime>Singleton</LifeTime>
<Options>
<FailInitializationTimeoutMilliseconds>100000</FailInitializationTimeoutMilliseconds>
<RetryTimeoutMilliseconds>100</RetryTimeoutMilliseconds>
</Options>
</IIndexingInitializer>
```

For more information about this setting, see the [developer documentation for Sitecore](#).

## 9.2.2 Upgrade the Sitecore Marketing Automation Engine service

1. In the Sitecore 9.0.2 rev. 180604 (OnPrem) `_xp0xconnect.scwdp\Content\Website\App_data\jobs\continuous\AutomationEngine` folder, copy all the files – not the folders – and paste them into the `xconnect\App_data\jobs\continuous\AutomationEngine` folder.

2. In the `sc.MarketingAutomation.ActivityTypes.xml` file, remove the `<MarketingAutomation.Activity.ConditionalPlanEntry>` node with this ID:

```
<Id>{2EAF2623-BE5F-4DB7-8A29-BF50423E01BD}</Id> ID
```

3. In the `sc.MarketingAutomation.Processors.xml` file, after the `</MarketingAutomation.Processing.LiveEventProcessor>` node, register the new `<MarketingAutomation.Processing.OperationResultCache>` service:

```
<!-- A cache used by processors to store intermediate results -->
<MarketingAutomation.Processing.OperationResultCache>
<Type>Sitecore.Xdb.MarketingAutomation.Cache.MemoryOperationResultCache,
Sitecore.Xdb.MarketingAutomation</Type>
<As>Sitecore.Xdb.MarketingAutomation.Core.Cache.IOperationResultCache,
Sitecore.Xdb.MarketingAutomation.Core</As>
<LifeTime>Singleton</LifeTime>
<Options>
<!-- The duration entries will live in the cache for -->
<EntryLifetime>0:00:30</EntryLifetime>
</Options>
</MarketingAutomation.Processing.OperationResultCache>
```

4. In the Sitecore 9.0.2 rev. 180604 (OnPrem) `_xp0xconnect.scwdp\Content\Website\App_data\jobs\continuous\AutomationEngine\App_Data\Config\sitecore\Messaging` folder, copy the `sc.XConnect.TransientFaultHandling.xml` file and paste it into the `xconnect\jobs\continuous\AutomationEngine\App_Data\Config\sitecore\Messaging` folder.
5. In the Sitecore 9.0.2 rev. 180604 (OnPrem) `_xp0xconnect.scwdp\Content\Website\App_data\jobs\continuous\AutomationEngine\App_data\Diagnostics` folder, copy all the files and paste them into the `xconnect\App_data\jobs\continuous\AutomationEngine\App_data\Diagnostics` folder.

## 9.2.3 Resuming Services

1. Recycle the IIS XConnect site Application Pool.
2. Start the *Sitecore XConnect Search Indexer* Windows service
3. Start *Sitecore Marketing Automation Engine* Windows service
4. [Rebuild the XConnect indexes.](#)

## Chapter 10

### Post-upgrade Steps

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

This chapter contains the following sections:

- General Maintenance
- xDB Maintenance
- Upgrading from Sitecore XP 8.1 – Sitecore XP 8.2.7
- Upgrading from Sitecore XP 9.0 or Sitecore 9.0 Update 1

## 10.1 General Maintenance

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

- Clear the browser cache.
- Republish your website.  
You must publish the entire site, including system items, templates, and so on, to every publishing target.
- Rebuild the search indexes.
  - 1) On the **Sitecore Launchpad**, click **Control Panel**, and in the **Indexing** section, click **Indexing manager**.
  - 2) In the **Indexing Manager** dialog box, click **Select all**, and then click **Rebuild**.
- Rebuild the *Link* database in the *Master* and *Core* databases.
  - 1) On the **Sitecore Launchpad**, click **Control Panel**, and in the **Database** section, click **Rebuild Link Databases**.
  - 2) Select the *Master* and *Core* databases and then click **Rebuild**.
- Import client languages.  
If you are using a client language other than English, download and import the latest translations from the *Client Translations* section of the [release page](#).

### 10.1.1 Cache Configuration

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. This topic is *only* valid if you have upgraded from any version of Sitecore XP 8.1 to Sitecore XP 9.0.2.

For more information about cache configuration, see the [Cache Configuration Reference](#) document on the Sitecore Developer Network.

## 10.2 xDB Maintenance

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

### 10.2.1 Enable xDB and Experience Analytics

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

To enable xDB and Experience Analytics:

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

### 10.2.2 Using EXM

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

EXM is enabled by default.

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

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

The following values are supported:

- yes
- no

The default value is yes.

#### Important

EXM only works when xDB is enabled.

### Define the Email Delivery Provider

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

To define the email delivery provider:

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

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

The following values are supported:

- `CustomSmtplib`
- `EmailCloud`

The default value is `CustomSmtplib`.

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

### 10.2.3 Redeploy the Marketing Definitions

To redeploy the marketing definitions:

1. On the **Sitecore Launchpad**, click **Control Panel, Analytics**, and then click **Deploy Marketing Definitions**.

2. In the **Deploy marketing definitions** dialog box, select all the definitions and taxonomies and click **Deploy**.

## Note

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

## 10.2.4 Upgrade the Session Database

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

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

## 10.3 Upgrading from Sitecore XP 8.1 – Sitecore XP 8.2.7

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

### 10.3.1 Delete Unwanted Marketing Definitions and Marketing Taxonomies

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

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

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

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

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

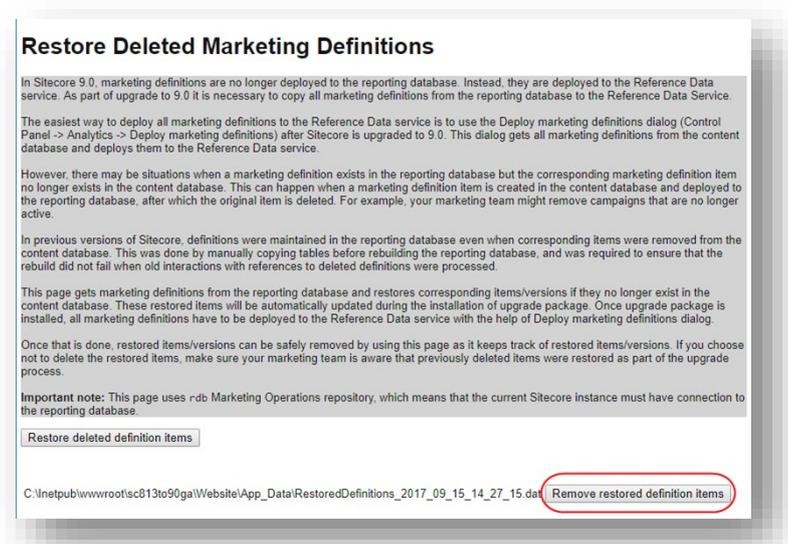
- In a text editor, in the `\App_Data` folder, open the `RestoredTaxonomies_{timestamp}.dat` file.

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

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

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

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

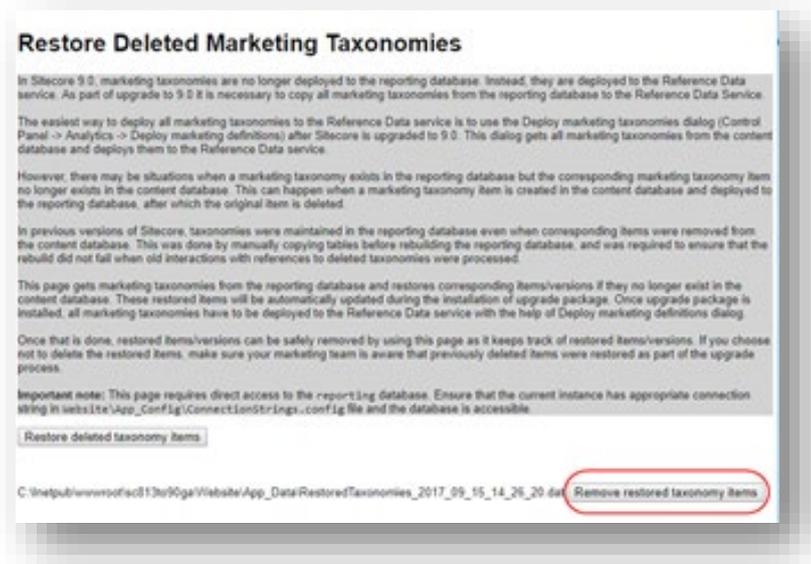


To delete unwanted marketing taxonomy items from the *Master* database:

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

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

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



## 10.3.2 Update Customizations for Reporting

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

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

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

## 10.3.3 Upgrade Experience Analytics

In Sitecore 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 an Experience Analytics Report](#).

## 10.3.4 Upgrade Path Analyzer

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

### Update Custom Maps

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

### Redeploy the Path Analyzer Maps

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

To redeploy the Path Analyzer maps:

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

## 10.4 Upgrading from Sitecore XP 9.0 or Sitecore 9.0 Update 1

If you are upgrading from Sitecore XP 9.0 or Sitecore XP 9.0 Update 1, you must perform the tasks described in this section.

### 10.4.1 Upgrade the Contact Merge Data

In Sitecore XP 9.0 Update 2, the contact merge behavior has changed. In Sitecore XP 9.0 Update 2, 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 Sitecore XP 9.0.2 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.

## Chapter 11

### Upgrade Multiple Instances

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

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

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

This chapter contains the following section:

- Upgrade a Scaled Environment With Multiple Instances

## 11.1 Upgrade a Scaled Environment With Multiple Instances

To upgrade a scaled environment with multiple instances, for example, a scaled environment that consists of Content Delivery, Content Management, Processing and Reporting instances that are connected to the same databases – *Core*, *Master*, *Web*, and *Reporting*, you must follow all the steps described in this guide, with the following differences:

### Content Management

Complete all the steps in this guide.

### Content Delivery, Processing, and Reporting

Complete all the steps in this guide, with the following differences:

- In the chapter *Prepare for the Upgrade*, you must only follow the steps in the section *Disable the Sitecore Experience Database*.
- In the section *Use the Update Installation Wizard to Install the Upgrade Package*:
  - In the **Analyze the package** dialog box, in the **Advanced options** section, clear the **Install items** check box.
- In the section *Specify the Server Role and the Search Provider*:
  - Set the `role:define` setting to either the ContentDelivery, Processing, or Reporting instance.
- In the chapter *Upgrade Web Forms for Marketers*:
  - In the **Analyze the package** dialog box, in the **Advanced options** section, clear the **Install items** check box.
- You must also skip all of the steps in the chapter *Post-upgrade Steps*.

### Session State Provider

To upgrade the MongoDB Session State Provider, you must specify the `uuidRepresentation` parameter in the connection string, for example:

```
mongodb://localhost/session?uuidRepresentation=standard
```

### Scaled XConnect Environment

If you are running Sitecore XP 9.0 Initial Release with the scaled XP1 XConnect topology, you must perform all the upgrade steps described in the chapter *Upgrade xConnect from Sitecore XP 9.0 Initial Release to Sitecore XP 9.0 Update 2* in alignment with the instructions in the [Sitecore Scaling and Architecture Guide](#).

If you are running Sitecore XP 9.0 Update 1 with the scaled XP1 XConnect topology, you must perform all the upgrade steps described in the chapter *Upgrade xConnect from Sitecore XP 9.0.1 to Sitecore XP 9.0 Update 2* in alignment with the instructions in the [Sitecore Scaling and Architecture Guide](#).

The Sitecore Scaling and Architecture Guide tells you the path to 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.

For more information, see the [Sitecore Scaling and Architecture Guide](#).

## Chapter 12

### Migrate xDB data

**Important**

If you are upgrading from Sitecore XP 8.1 – Sitecore XP 8.2 Update 7 to Sitecore XP 9.0 Update 2, you must perform this step.

If you are upgrading from Sitecore XP 9.0 or Sitecore XP 9.0.1 to Sitecore XP 9.0 Update 2, skip this step.

In Sitecore XP 9.0, we introduced a new data model for the Experience Database (xDB). This means that you must convert all data in the MongoDB database to the new format. Use the xDB Data Migration Tool to perform this data conversion.

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

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

This chapter contains the following section:

- Use the xDB Data Migration Tool to Migrate xDB data

## 12.1 Use the xDB Data Migration Tool to Migrate xDB data

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

To deploy the new *Reporting* database:

1. Deploy the `Sitecore.Analytics.dacpac`. This file is located in the `Sitecore 9.0.2 rev. 180604 (upgrade files).zip` package, in the `\Databases` folder.
2. In the `ConnectionStrings.config` file of your Sitecore instance, replace the name of old reporting database with the name of the new one. For example:

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

### 12.1.1 Migrate the xDB Data

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

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

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

## Chapter 13

### Breaking Changes from EXM 3.5

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

This chapter contains the following sections:

- Breaking Changes

## 13.1 Breaking Changes

There are over 1,000 breaking changes, including:

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

### 13.1.1 Factories Replaced

The following factories have been removed:

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

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

You can find most of the new factories in:

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

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

### 13.1.2 Client API and Messaging Bus

The following files have been removed:

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

The new messaging bus, shipped with 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.