

# Sitecore Experience Platform Upgrade Guide

Sitecore Experience Platform 9.3.0

*Upgrade to Sitecore XP 9.3.0 from Sitecore XP 9.0.2 or later*

## Table of Contents

Chapter 1	Introduction.....	4
Chapter 2	Preparing to Upgrade Sitecore XP .....	5
2.1	Files and scripts for upgrading.....	6
2.2	Preparing for the upgrade .....	7
2.2.1	Obtain and set up certificates for SSL.....	7
2.2.2	Stop active content tests .....	7
2.2.3	Upgrade the databases .....	7
2.2.4	Disable the Sitecore modules .....	8
2.2.5	Disable Web Forms for Marketers .....	8
2.2.6	Sitecore Experience Accelerator.....	8
2.2.7	Disable active Sitecore services.....	9
2.2.8	Disable custom DLLs and configuration files .....	9
2.2.9	Add the Security Connection String.....	9
Chapter 3	Using the Sitecore Update Center .....	10
3.1	Sitecore Update Center .....	11
3.1.1	Update mode .....	12
3.1.2	Configuration .....	12
3.2	Using the Sitecore Update Center .....	13
Chapter 4	Upgrade the Sitecore xConnect Server .....	14
4.1	Prepare to upgrade the xConnect Server .....	15
4.1.1	Prerequisites for Upgrading xConnect .....	15
4.1.2	Backup your customizations.....	15
4.1.3	Stop Windows services .....	16
4.2	Upgrade the xDB databases.....	17
4.2.1	Marketing Automation database.....	17
4.2.2	xDB Reference Data database .....	17
4.2.3	Messaging database.....	17
4.2.4	xDB Collection database on SQL Server.....	17
4.2.5	xDB Collection database on MongoDB .....	18
4.2.6	Processing Engine databases.....	19
4.2.7	Processing Tasks database .....	19
4.2.8	Processing Pools database.....	19
4.3	Update the files .....	20
4.3.1	Add new connection strings.....	21
4.3.2	Reconfigure xConnect Search Indexer service .....	21
4.3.3	Reconfigure MA Engine service .....	22
4.3.4	Reconfigure the SQL Provider for the Collection database .....	22
4.3.5	Restore customizations .....	22
4.4	Install and configure Sitecore Cortex Processing Engine .....	23
4.4.1	Install and configure Microsoft Machine Learning Server .....	24
Chapter 5	Upgrading Sitecore XP .....	25
5.1	Using the Update Center to upgrade to Sitecore XP 9.3.0 .....	26
5.1.1	Install the Upgrade Package .....	26
5.2	Troubleshooting .....	27
Chapter 6	Configure Sitecore .....	28
6.1	Review changes and update configuration files .....	29
6.1.1	Helix folder Structure.....	29

6.1.2	Configuration file changes .....	29
6.1.3	Patch files with customizations .....	29
6.1.4	Review the Web.config customizations .....	30
6.1.5	Search provider settings.....	30
6.1.6	Server role settings.....	31
6.2	Upgrading Solr.....	33
Chapter 7	Post-upgrade Steps .....	34
7.1	General maintenance .....	35
7.1.1	Upgrade the Session database.....	35
7.1.2	Update personalization rules.....	36
7.2	Enable telemetry reporting in a production environment.....	37
7.3	xDB Maintenance.....	39
7.3.1	Resume the xConnect services.....	39
7.3.2	Exiting Update Mode.....	39
7.3.3	Using Email Experience Manager (EXM).....	40
7.3.4	Redeploy the marketing definitions.....	40
7.3.5	Update Sitecore.Analytics.ExcludeRobots.config file.....	41
7.3.6	Synchronizing Experience Analytics with Reference Data .....	41
7.3.7	Remove Deprecated Path Analyzer Maps from the Reporting Database.....	41
7.3.8	Upgrading from Sitecore 9.0.2.....	42
7.3.9	Rebuild the Reporting database.....	43
7.4	Recompile your solution .....	44
Chapter 8	Sitecore Identity Server.....	45
8.1	Install Sitecore Identity Server.....	46
8.1.1	Configure Sitecore Identity Client.....	47
8.2	Upgrade Sitecore Identity Server .....	48
Chapter 9	Upgrade Multiple Instances .....	49
9.1	Upgrade a scaled environment with multiple instances .....	50
Chapter 10	Appendix A: The xConnect Application Preparation Script.....	51
Chapter 11	Appendix B: Generating shard keys and missed percentile values.....	52
11.1	Run the Sitecore Xdb Collection Update tool .....	53
11.1.1	Run the tool.....	53
11.1.2	Update operations .....	53
11.1.3	Tool arguments:.....	54

*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 © 2001-2021 Sitecore. All rights reserved.*

# Chapter 1

## Introduction

This guide describes how to upgrade from Sitecore Experience Platform 9.0.2 or later to Sitecore Experience Platform 9.3.0 and applies only to the on-premise version.

If you are upgrading from Sitecore XP 9.0.1 or earlier, use the *Upgrade to Sitecore XP 9.3.0 from Sitecore XP 8.1.0 - 9.0.1 guide* available at the [Sitecore Downloads](#) site.

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

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

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

## Chapter 2

# Preparing to Upgrade Sitecore XP

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

Before you upgrade to Sitecore XP 9.3.0, you should ensure that you meet the software and hardware requirements for this version of Sitecore XP. For more information about the requirements, see the [Sitecore XP 9.3.0 Installation Guide](#).

**Note**

With Sitecore XP 9.3, you can no longer use Lucene as your search provider.

This chapter contains the following sections:

- Files and scripts for upgrading
- Preparing for the upgrade

## 2.1 Files and scripts for upgrading

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

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

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

To upgrade from Sitecore XP 9.1.0 or XP 9.1.1, you require the following file:

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

To upgrade from Sitecore XP 9.0.2 - 9.2.0, you also require the following file which is listed on the download site as *Configuration files for upgrade*:

- Sitecore 9.3.0. rev. 003498 (upgrade files).zip

This file contains:

- o Database Upgrade Script.zip

This file contains several files, including the database upgrade scripts.

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

## 2.2 Preparing for the upgrade

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

### 2.2.1 Obtain and set up certificates for SSL

#### Note

If you already use HTTPS, skip this section.

With the release of Sitecore XP 9.3.0, all communication with the Content Management server occurs over HTTPS. HTTPS requires that you obtain and set up certificates for the Secure Sockets Layer (SSL) before you upgrade the platform.

For more information about how to set up SSL in IIS, see [Microsoft's documentation](#).

For more information about how to install the server certificates, see the [Installation Guide](#).

### 2.2.2 Stop active content tests

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

### 2.2.3 Upgrade the databases

The scripts that you use to upgrade the databases are available in the `Database Upgrade Script.zip` file.

To upgrade the databases:

1. Run the `CMS_core.sql` script for the *Core* database.
2. Run the `CMS_core_master_web9x.sql` script for the *Core*, *Master*, and *Web* databases.
3. Run the `SXP_reporting.sql` script for the *Reporting* database.

Among other things, this script adds two new columns – *Converted* and *FirstImpressionCount* – to some of the reporting database artifacts. These columns are used to increase the accuracy of conversion and bounce rate calculations. The script sets the default value of the columns to 0.

If you want to set different default values in these columns, do one of the following:

- Retain the previous conversion and bounce rate calculations by uncommenting the lines that come after the following comments before running the script:

For Conversion Rates:

```
--Uncomment the script below if you want to retain the previous conversion rate results.
```

For Bounce Rates:

```
--Uncomment the script below if you want to retain the previous bounce rate results.
```

- Retrieve the correct conversion and bounce rate calculations by keeping the *Converted* and *FirstImpressionCount* to 0.

4. Run the `SXP_reporting_migrate_experience_optimization_data.sql` for the *Reporting* database.

This script restructures the data in the *Fact\_PageViews* and *Fact\_Visits* tables.

5. If you are upgrading from Sitecore XP 9.0.2, run the `SXP_exm_master_dispatchqueue.sql` script for the *EXM.Master* database.
6. If you are upgrading from Sitecore XP 9.1.0 or earlier, run the `SXP_exm_master_suppressions.sql` script for the *EXM.Master* database.
7. If you are upgrading from Sitecore XP 9.1.0 or earlier, run the `SXP_exm_master_save_suppressions.sql` script for the *EXM.Master* database.
8. If you are upgrading from Sitecore XP 9.1.0 or earlier, run the `SXP_exm_master_update_suppressions.sql` script for the *EXM.Master* database.
9. Run the `SXP_experienceforms_storage.sql` for the *ExperienceForms* database.
10. Run the `SXP_experienceforms_filestorage.sql` for the *ExperienceForms* database.

## 2.2.4 Disable the Sitecore modules

You must disable any Sitecore modules that are in use.

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

## 2.2.5 Disable Web Forms for Marketers

### Important

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

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

To disable the Web Forms for Marketers module:

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

## 2.2.6 Sitecore Experience Accelerator

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

## 2.2.7 Disable active Sitecore services

Before upgrading, you must disable some active services.

To disable the active services:

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

If you are upgrading from Sitecore XP 9.0.2, these files are in the root `Website` folder.

2. In the `App_Config\Sitecore\LayoutService\` folder, delete the `Sitecore.LayoutService.Jss.config` file.
3. In the `App_Config\Sitecore\Services.GraphQL\` folder, delete the `zSitecore.JavaScriptServices.GraphQL.config` file.

## 2.2.8 Disable custom DLLs and configuration files

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

## 2.2.9 Add the Security Connection String

### Important

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

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

Before you upgrade to Sitecore XP 9.3.0, you must add the *security* connection string.

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

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

## Chapter 3

# Using the Sitecore Update Center

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

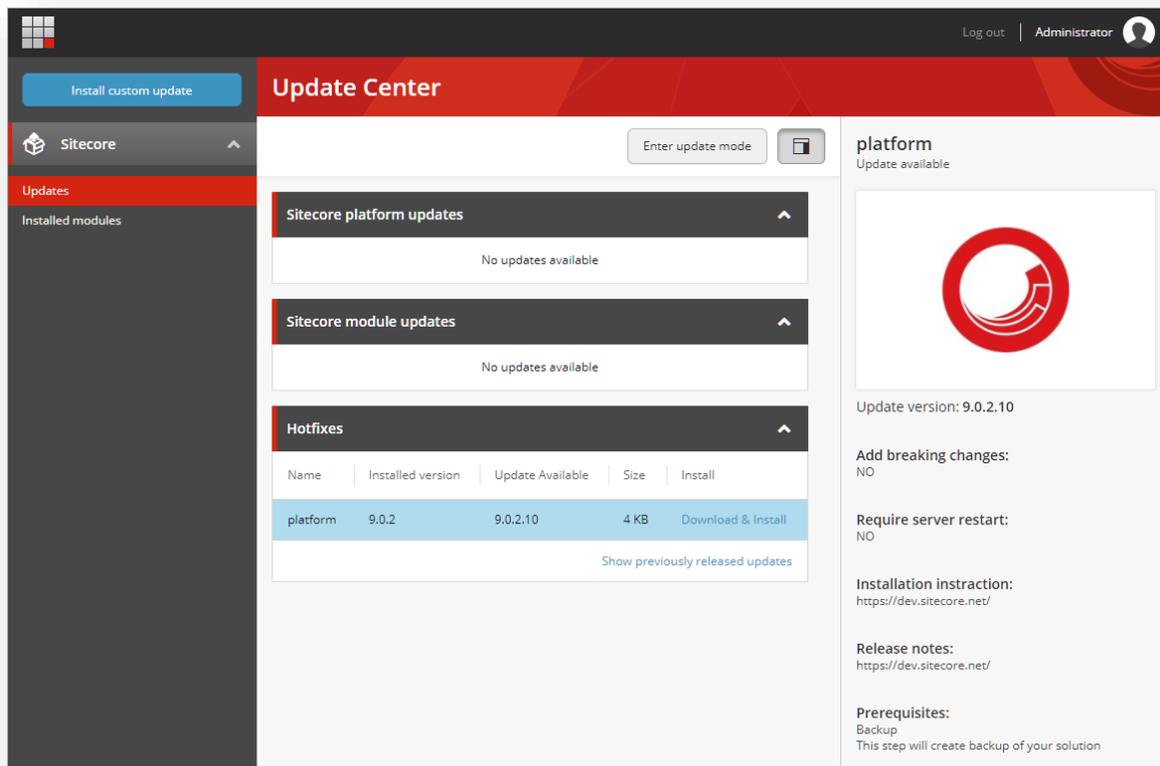
This chapter contains the following sections:

- Sitecore Update Center
- Using the Sitecore Update Center

### 3.1 Sitecore Update Center

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

You open Update Center from the Sitecore Dashboard:



The left pane, contains three options:

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

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

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

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

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

### 3.1.1 Update mode

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

The Sitecore configuration is divided into four layers:

- Sitecore
- Modules
- Custom
- Environment

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

The `Update` folder contains these configuration files:

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

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

### 3.1.2 Configuration

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

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

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

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

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

## 3.2 Using the Sitecore Update Center

The Update Center uses the Update Installation wizard to install upgrade packages.

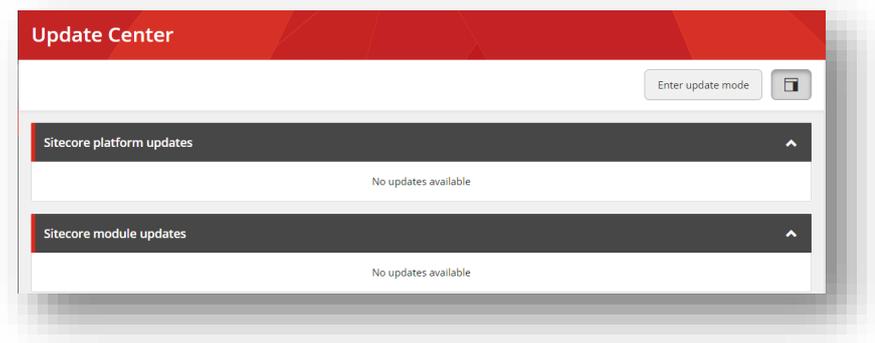
If you are upgrading from Sitecore XP 9.1.1 or earlier, you must update the Update Installation Wizard before you use the Update Center to install the Sitecore XP upgrade package.

To update the Update Installation Wizard:

1. On the **Sitecore Launchpad**, click **Control Panel**, and in the **Administration** section, click **Install a package**.
2. Do one of the following:
  - If you are upgrading from Sitecore XP 9.1.0 or XP 9.1.1, install the Sitecore Update Installation Wizard 4.0.1 rev.00153 for 9.1.zip package.
  - If you are upgrading from Sitecore XP 9.0.2, install the Sitecore Update Installation Wizard 4.0.1 rev.00153 for 8.x-9.0.zip package.

To enter update mode:

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



### Note

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

## Chapter 4

# Upgrade the Sitecore xConnect Server

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

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

- Sitecore XP 9.0.2
- Sitecore XP 9.1.0 (Initial Release that introduced the Sitecore Cortex™ Processing Engine role)
- Sitecore XP 9.1.1
- Sitecore XP 9.2.0

### **Important**

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

This chapter contains the following sections:

- Prepare to upgrade the xConnect Server
- Upgrade the xDB databases
- Update the files
- Install and configure Sitecore Cortex Processing Engine
- Configure the search provider
- Azure Key Vault

## 4.1 Prepare to upgrade the xConnect Server

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

### Note

If you are going to switch from SQL Server to MongoDB as part of the upgrade, you must configure the MongoDB provider for xConnect. For information about how to configure the MongoDB provider for xConnect, see the installation guide for [Sitecore XP 9.3.0](#).

### 4.1.1 Prerequisites for Upgrading xConnect

Before you upgrade xConnect:

1. [Download](#) the following files:
  - o Sitecore 9.3.0 rev. 003498 (WDP XP0 packages).zip.  
In the *Download options for On Premises deployment* section, this file is called *Packages for XP Single*.
  - o Sitecore 9.3.0 rev. 003498 (upgrade files).zip.  
In the *Upgrade options* section, this file is called *Configuration files for upgrade*. You might have already downloaded this file if you followed the procedures in the *Files and scripts for upgrading* section.
2. Unpack the Sitecore 9.3.0 rev. 003498 (WDP XP0 packages).zip file and then unpack the Sitecore 9.3.0 rev. 003498 (OnPrem)\_xp0xconnect.scwdp.zip file.
3. Unpack the Sitecore 9.3.0 rev. 003498 (upgrade files).zip file and then unpack the Database Upgrade Script.zip file. This file contains several database upgrade scripts.  
You might have already unpacked this file if you followed the steps in the section *Files and scripts for upgrading*.
4. If you have any functionality that accesses the contact/interaction data extraction process from outside of xConnect, you must stop it. You can restart it when the upgrade is completed. An example of this kind of functionality is rebuilding the *Reporting* database.

### Important

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

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

### 4.1.2 Backup your customizations

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

- Configuration files or configuration file patches
- Custom xConnect model JSON files

- Custom DLL files

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

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

### 4.1.3 Stop Windows services

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

Stop the following Windows services:

- Sitecore XConnect Search Indexer
- Sitecore Marketing Automation Engine
- Sitecore Cortex Processing Engine - if you are upgrading from Sitecore XP 9.1.0 or later.

Before you start the upgrade, we recommend that you:

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

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

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

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

## 4.2 Upgrade the xDB databases

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

### 4.2.1 Marketing Automation database

#### Important

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

To upgrade the *Marketing Automation* database:

- Run the `SXP_marketingautomation.sql` script for the *Marketing Automation* database.

### 4.2.2 xDB Reference Data database

If you are upgrading from Sitecore XP 9.0.2 or later, you must upgrade the *xDB Reference Data* database. We recommend that you make a copy of the *xDB Reference Data* database before completing this procedure.

To upgrade the *xDB Reference Data* database:

- Run the `SXP_referencedata_Part1.sql` script.
  - If you receive the following message:  
*# definitions cannot be migrated. Length of # definition monikers exceeds 300 characters. Consider truncate, change or remove them. Upgrade stopped. Fix all the issues and run this script again. Upgrade (part 1) has not been completed.*  
  
Fix all the issues as described in this message and run the `SXP_referencedata_Part1.sql` script again.  
  
Repeat this step until you receive the *"Upgrade (part 1) has been done successfully"* message.
- Run the `SXP_referencedata_Part2.sql` script.

### 4.2.3 Messaging database

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

### 4.2.4 xDB Collection database on SQL Server

#### Important

If the *xDB Collection* database is on SQL Server follow the procedures described in this section. Otherwise, skip this section.

#### Upgrade the xDB Collection database - Shard Map Manager

Follow this procedure if you are going to apply scaling to the *xDB Collection* database.

Run the `SXP_collection_smm.sql` script for the *Shard Map Manager* database to register the schemas for shard maps. This is required for the *Split/Merge Azure Service* that is used for data rebalancing and horizontal scaling.

## Upgrade the xDB Collection database – Shards

Regardless of whether or not the *Always Encrypted* feature is configured for the *xDB Collection* database:

- Run the `SXP_collection_GrantPermissions.sql` script or run the following SQL script for every least privileged user in every shard.

You must specify the user name in the script.

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

### With Always Encrypted

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

1. Run the `SXP_collection_AE_Part1.sql` script for every shard.
2. In the new `[UnlockContactIdentifiersIndex_Staging]` table, configure [deterministic encryption](#) for the `[Identifier]` and `[Source]` columns, for every shard.  
To simplify the configuration process, SQL Server Management Studio provides an [Encryption Wizard](#).
3. Run the `SXP_collection_AE_Part2.sql` script for every shard.

### Without Always Encrypted

If the *Always Encrypted* feature is not configured for the *xDB Collection* database:

- Run the `SXP_collection_Part1.sql` script for every shard.

## SQL Azure

If your *xDB Collection* database is deployed on SQL Azure, you can use the split/merge service to distribute analytics data and your collection databases.

To use the split/merge service:

1. Run the *Sitecore Xdb Collection Update* tool to generate shard keys and any missing percentile values from the record IDs. For more information, see *Appendix B: Generating shard keys and missed percentile values*.
2. After you have run the *Sitecore Xdb Collection Update* tool, run the `SXP_collection_Part2.sql` script for every shard.

## 4.2.5 xDB Collection database on MongoDB

### Important

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

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

The installation guide for Sitecore XP 9.3.0 describes how to configure the MongoDB provider for xConnect.

To continue using the MongoDB xDB Collection database provider with Sitecore XP 9.3.0, you must upgrade your existing MongoDB instance to version 4.0.5 – 4.0.10.

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

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

To build indexes efficiently, follow the suggestions in the article [Index Build Operations on a Populated Collection](#).

## 4.2.6 Processing Engine databases

### Important

If you are upgrading from Sitecore XP 9.1.0 or later, you must upgrade the *Processing Engine Task* and *Processing Engine Storage* databases.

To upgrade the *Processing Engine Tasks* database:

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

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

To upgrade the *Processing Engine Storage* database:

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

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

## 4.2.7 Processing Tasks database

If you are upgrading from Sitecore XP 9.0.2 or later, you must upgrade the *Processing Tasks* database:

- Run the `SXP_processing_tasks.sql` script for the *Processing Tasks* database.

## 4.2.8 Processing Pools database

If upgrading from Sitecore XP 9.0.2 - 9.1.1, you must upgrade the *Xdb Processing Pools* database.

To upgrade the *Xdb Processing Pools* database:

- Run the `SXP_processing_pools.sql` script for the *Xdb Processing Pools* database.

## 4.3 Update the files

Update the files on the xConnect instance.

### Note

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

To update the files:

1. Delete all the files under the following folders, but leave the folders themselves to ensure that the file permissions are preserved:

- o \Bin
- o \App\_Data\collectiondeployment
- o \App\_Data\Config
- o \App\_Data\Copyrights
- o \App\_Data\Diagnostics
- o \App\_Data\Models
- o \App\_Data\solrcommands
- o \App\_Data\jobs\continuous\AutomationEngine
- o \App\_Data\jobs\continuous\AutomationEngine\AppData\Config
- o \App\_Data\jobs\continuous\AutomationEngine\AppData\Diagnostics
- o \App\_Data\jobs\continuous\IndexWorker
- o \App\_Data\jobs\continuous\IndexWorker\Copyrights
- o \App\_Data\jobs\continuous\IndexWorker\AppData\Config
- o \App\_Data\jobs\continuous\IndexWorker\AppData\Diagnostics
- o \App\_Data\jobs\continuous\IndexWorker\AppData\Models
- o \App\_Data\jobs\continuous\ProcessingEngine
- o \App\_Data\jobs\continuous\ProcessingEngine\AppData\Config
- o \App\_Data\jobs\continuous\ProcessingEngine\AppData\Diagnostics
- o \App\_Data\jobs\continuous\ProcessingEngine\AppData\Models

### Note

If you are upgrading from SC XP 9.0.2, you must create the \App\_Data\jobs\continuous\ProcessingEngine folder.

2. In the \App\_Data\jobs\continuous\IndexWorker folder, delete the following language folders if they exist:

- o \de-DE
- o \es-ES
- o \fr-FR
- o \it-IT
- o \ja-JP
- o \ko-KR

- o \pt-BR
  - o \ru-RU
  - o \zh-CN
  - o \zh-TW
3. Copy all the files and folders from the Sitecore 9.3.0 rev. 003498 (OnPrem)\_xp0xconnect.scwdp\Content\Website folder that you unpacked earlier, except for the following folders:
- o \App\_Config
  - o \App\_Data\jobs\continuous\AutomationEngine\App\_Config
  - o \App\_Data\jobs\continuous\IndexWorker\App\_Config
  - o \App\_Data\jobs\continuous\ProcessingEngine\App\_Config
- If you are upgrading from SC XP 9.0.2 you must copy this folder to get all the files for the processing engine.
- o \App\_Data\logs

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

### 4.3.1 Add new connection strings

#### Important

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

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

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

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

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

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

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

### 4.3.2 Reconfigure xConnect Search Indexer service

In Sitecore XP 9.3.0, `xConnectSearchIndexer.exe` was renamed to `Sitecore.XConnectSearchIndexer.exe`.

To ensure that your xConnect Search Indexer service points to the executable with the new name, run the following command:

```
sc config <your xConnect Search Indexer service name> binpath= "C:\inetpub\wwwroot\<your
xConnect instance
name>\App_Data\jobs\continuous\IndexWorker\Sitecore.XConnectSearchIndexer.exe -service"
```

### 4.3.3 Reconfigure MA Engine service

In Sitecore 9.3.0, `maengine.exe` was renamed to `Sitecore.MAEngine.exe`.

To ensure that your *MA Engine* service points to the executable with the new name, run the following command:

```
sc config <your MA Engine service name> binpath= "C:\inetpub\wwwroot\<your xConnect instance name>\App_Data\jobs\continuous\AutomationEngine\Sitecore.MAEngine.exe -service"
```

### 4.3.4 Reconfigure the SQL Provider for the Collection database

On SQL Server, the *Always Encrypted* feature is used to encrypt sensitive data.

In Sitecore XP 9.3.0, xConnect is configured to use table-valued parameters on SQL Server. However, table-valued parameters are not compatible with the *Always Encrypted* feature.

If you want to continue using the *Always Encrypted* feature, you must configure your installation to use staging tables instead of table-valued parameters.

If the *Collection* database is configured to use the *Always Encrypted* feature and you want to continue using this feature, in the `xconnect\App_Data\Config\Sitecore\Collection\sc.Xdb.Collection.Data.Sql.xml` configuration file, replace the `Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy` sections with

```
<Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.IContactsExecutionStrategy>
  <Type>Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.ContactsStagingTableExecutionStrategy, Sitecore.Xdb.Collection.Data.SqlServer</Type>
  <As>Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.IContactsExecutionStrategy, Sitecore.Xdb.Collection.Data.SqlServer</As>
  <LifeTime>Singleton</LifeTime>
</Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.IContactsExecutionStrategy>
<Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.IInteractionsExecutionStrategy>
  <Type>Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.InteractionsStagingTableExecutionStrategy, Sitecore.Xdb.Collection.Data.SqlServer</Type>
  <As>Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.IInteractionsExecutionStrategy, Sitecore.Xdb.Collection.Data.SqlServer</As>
  <LifeTime>Singleton</LifeTime>
</Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.IInteractionsExecutionStrategy>
<Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.IDeviceProfilesExecutionStrategy>
  <Type>Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.DeviceProfilesStagingTableExecutionStrategy, Sitecore.Xdb.Collection.Data.SqlServer</Type>
  <As>Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.IDeviceProfilesExecutionStrategy, Sitecore.Xdb.Collection.Data.SqlServer</As>
  <LifeTime>Singleton</LifeTime>
</Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.IDeviceProfilesExecutionStrategy>
```

### 4.3.5 Restore customizations

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

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

## 4.4 Install and configure Sitecore Cortex Processing Engine

You have already installed the files for the Cortex Processing Engine. Now you must deploy the Processing Engine databases and configure the Processing Engine service.

To deploy the databases, in the Sitecore 9.3.0 rev. 003498

(OnPrem)\_xp0xconnect.scwdp\Content folder, find and deploy the following files to SQL server:

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

To configure the Processing Engine:

1. In the `instanceName.xconnect\App_data\jobs\continuous\ProcessingEngine\App_Config\ConnectionStrings.config` file, update the following connection strings:
  - o `processing.engine.storage`
  - o `processing.engine.tasks`
  - o `processing.webapi.blob`
  - o `processing.webapi.table`
  - o `xconnect.collection`
  - o `xconnect.collection.certificate`
  - o `xconnect.configuration`
  - o `xconnect.configuration.certificate`
  - o `xconnect.search`
  - o `xconnect.search.certificate`
  - o `messaging`
  - o `reporting`

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

### Note

The `processing.webapi.blob` and `processing.webapi.table` APIs are deployed with your xConnect instance and should point to your xConnect endpoint for example: <https://instanceName.xconnect>

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

2. In the `xconnect\App_data\jobs\continuous\ProcessingEngine\App_Data` folder, add a valid `license.xml` file
3. In the `xconnect\App_data\jobs\continuous\ProcessingEngine\App_Data` folder, create a `Logs` folder and grant the `NT AUTHORITY\LOCAL SERVICE` security group *Full control* permission to it.

4. To install the Sitecore Cortex Processing Engine to run as a Windows Service, run the Command Prompt with *Administrator* permissions, run the `sc create` command, and pass the appropriate parameters:

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

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

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

#### 4.4.1 Install and configure Microsoft Machine Learning Server

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

##### Note

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

For more information about how to install Microsoft Machine Learning Server see the [Sitecore Experience Platform 9.3.0 Installation Guide](#).

## Chapter 5

# Upgrading Sitecore XP

You use the Sitecore Update Center to upgrade the Sitecore platform and modules. Before you use the Update Center to upgrade to Sitecore XP 9.3.0, you must have all the prerequisite steps that are described in this guide completed.

**Note**

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

This chapter contains the following sections:

- Using the Update Center to upgrade to Sitecore XP 9.3.0
- Troubleshooting

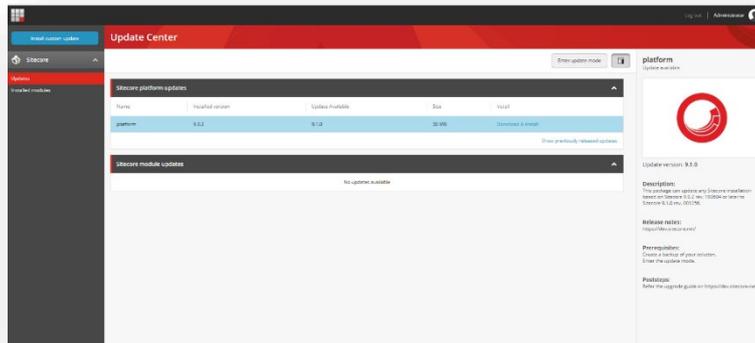
## 5.1 Using the Update Center to upgrade to Sitecore XP 9.3.0

### 5.1.1 Install the Upgrade Package

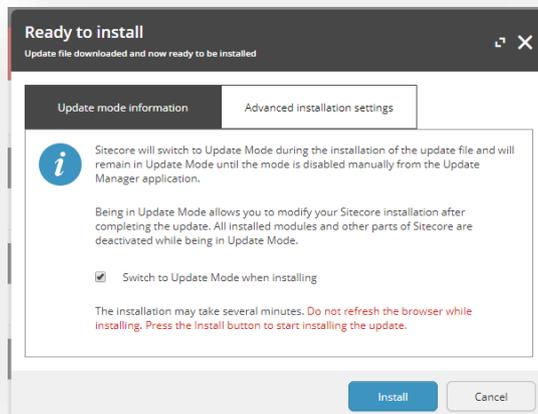
You can now use the Update Center to install the upgrade package. Before doing so, ensure you have followed the steps described in chapter *Preparing to Upgrade*.

To install the upgrade package:

1. On the Sitecore Launchpad, open the **Update Center**.



2. Select the Sitecore XP 9.3.0 upgrade package and then click **Download and Install**.
3. When the package is downloaded, in the **Ready to install** dialog box, click **Install**:



4. Wait until the package is installed.

After the upgrade package has been installed, you must configure Sitecore as described later in this document.

## 5.2 Troubleshooting

Some of the issues that can occur during the upgrade process.

### **I cannot see an update to Sitecore XP 9.3.0 in the Update Center**

Check the error message that appears in the Update Center.

### **When I click Install, the installation fails**

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

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

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

The update process is divided into three steps:

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

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

## Chapter 6

# Configure Sitecore

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

This chapter contains the following sections:

- Review changes and modify configuration files
- Upgrading Solr or SolrCloud

## 6.1 Review changes and update configuration files

You must review some configuration settings and make changes to some configuration files.

### 6.1.1 Helix folder Structure

#### Note

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

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

#### Note

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

The folders and files are loaded in the following order:

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

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

### 6.1.2 Configuration file changes

In Sitecore XP 9.3.0, some configuration files were changed.

The configuration of sites login, shell, and modules\_shell has been moved from the `Sitecore.Config` file to the new `App_Config\Sitecore\CMS.Core\Sitecore.Sites.config` file.

The configuration of the `Core` database, the `TaskDatabase` element, eventing, authentication, authorization, user management, roles, and access rights has been moved from the `Sitecore.Config` file to new configuration files in the `App_Config\Sitecore\CMS.Core\` folder.

### 6.1.3 Patch files with customizations

After you install 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.

### 6.1.4 Review the Web.config customizations

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

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

To enable your customizations:

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

**Note**

Any customizations in the `web.config` file are stored in the `Web.merged.config.disabled` file after the upgrade.

### 6.1.5 Search provider settings

Since Sitecore XP 9.0, the search provider is specified in the `Web.config` file. After upgrading to Sitecore 9.3.0 the search provider is automatically set to `Solr` in the updated `Web.config` file.

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

If your search provider before upgrading was:

- `Solr` – skip the rest of this section.
- `Azure` – modify the `Web.config` file and specify `Azure` as the search provider.
- `Lucene` – modify the `Web.config` file and specify `Solr` or `Azure` as the search provider. Support for `Lucene` was dropped in Sitecore XP 9.3.0.

To specify the search provider:

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

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

The following values are supported:

- Solr - default value since Sitecore XP 9.2.0.
- Azure - Azure Search

[Upgrading the Solr search provider](#) is described later in this document.

For more information about how to configure Azure Search, see <https://doc.sitecore.com/developers/93/platform-administration-and-architecture/en/configure-azure-search.html>

For more information about how to configure Azure Search for xConnect, see <https://doc.sitecore.com/developers/93/platform-administration-and-architecture/en/configure-azure-search-for-xconnect.html>

## 6.1.6 Server role settings

After you upgrade to Sitecore XP 9.3.0, the server role is automatically set to `Standalone` in the updated `Web.config` file. The information from the original version of the `Web.config` file that was used before the upgrade is saved as `Web.custom.config.disabled`. All your previous customizations are also stored in this file.

If the server role before the upgrade was:

- Standalone – skip the rest of this section.
- Any other setting – modify the `Web.config` file and re-set the server role to the value that you used before the upgrade.

### Note

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

To specify the server role:

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

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

The following server roles are supported:

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

- Standalone (default)

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

## Indexing sub-role

Sitecore XP 9.2.0 introduced a new *Indexing* sub-role to simplify the configuration of content indexing.

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

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

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

To specify the search provider:

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

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

The following values are supported:

- `Solr` - the default value is `Solr` since Sitecore XP 9.2.0.
- `Azure` - Azure Search

### Note

Support for the Lucene search provider has been removed in Sitecore XP 9.3.0.

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

## 6.2 Upgrading Solr

To continue using the Solr or SolrCloud search provider in Sitecore XP 9.3.0, you must upgrade your Solr environment to version 8.1.1.

When you upgrade to a new major version of Solr, the high-level process is:

1. Upgrade the Solr environment to the new version.

For more information, see [https://lucene.apache.org/solr/guide/8\\_1/upgrading-a-solr-cluster.html](https://lucene.apache.org/solr/guide/8_1/upgrading-a-solr-cluster.html).

2. Create new cores (Solr) or collections (SolrCloud) for Sitecore XP and xConnect.

### Note

Do not use the [Solr IndexUpgraderTool](#) to migrate your old cores or collections to. You must rebuild them and rewrite the data.

### Important

Sitecore XP 9.3.0 has introduced a new Solr index – the *Personalization* index. You must create this new index.

3. Configure the upgraded Sitecore XP and xConnect instances to use the new Solr deployment.
4. Rebuild all the indexes.

This is described in the post-upgrade steps.

### Content search

For more information about how to set up Solr for use with Sitecore, see <https://doc.sitecore.com/developers/93/platform-administration-and-architecture/en/walkthrough--setting-up-solr.html>.

For more information about how to set up SolrCloud for use with Sitecore, see <https://doc.sitecore.com/developers/93/platform-administration-and-architecture/en/walkthrough--setting-up-solrcloud.html>.

### xConnect search

For more information about how to set up Solr for use with xConnect, see

<https://doc.sitecore.com/developers/93/platform-administration-and-architecture/en/configure-the-solr-search-provider.html>

For more information about how to set up SolrCloud for use with xConnect, see

<https://doc.sitecore.com/developers/93/platform-administration-and-architecture/en/walkthrough--using-solrcloud-for-xconnect-search.html>

## Chapter 7

# 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
- Enable telemetry reporting in a production environment
- xDB Maintenance
- Recompile your solution

## 7.1 General maintenance

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

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

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

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

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

### 7.1.1 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.  
If you are using the MongoDB session state provider, drop the *Session* database.  
If you are using the SQL Server session state provider, replace the *Session* database with a clean one from the Sitecore XP installation package.  
If you are using the Redis cache, clean the cache using Redis Cache service "Console".
2. Start all the Sitecore XP instances.

## 7.1.2 Update personalization rules

### Note

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

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

- Campaign was Triggered
- Entry Page
- Day of Week
- Device Type
- Matches Pattern

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

The following table lists the changes:

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

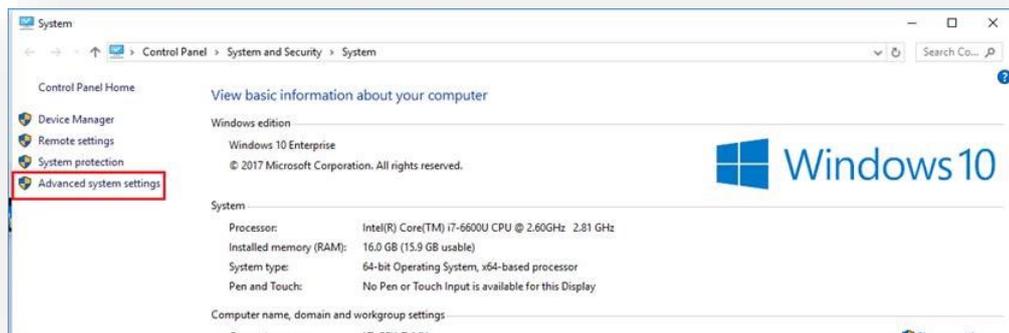
## 7.2 Enable telemetry reporting in a production environment

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

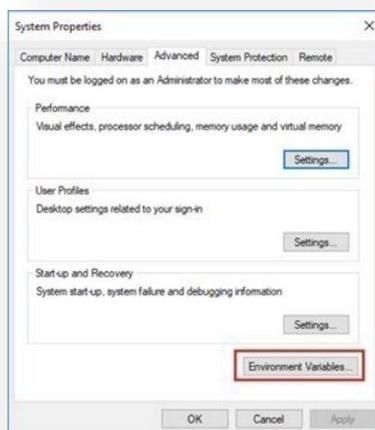
The telemetry system assumes Sitecore is running in a non-production environment by default. When you set up a production environment, you must therefore define a system variable to report the environment type.

To define a system variable:

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

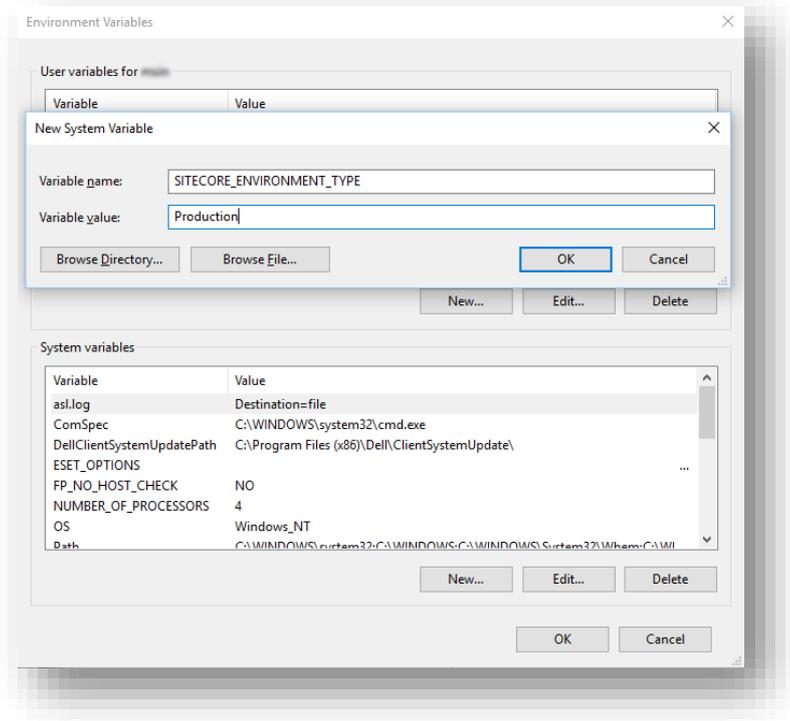


2. In the **System Properties** dialog, click **Environment Variables**.



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

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



5. Restart the computer.

## 7.3 xDB Maintenance

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

### 7.3.1 Resume the xConnect services

You must start the xConnect services that you stopped in the *Upgrade Sitecore xConnect* chapter. You must start these services before you enable xDB or exit Update Mode in the Upgrade Center.

To start the xConnect services:

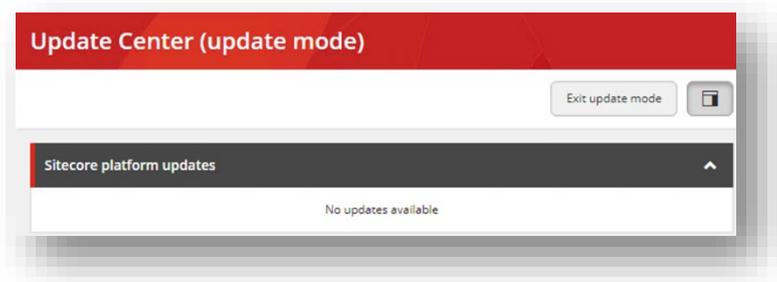
1. Recycle the IIS xConnect site Application Pool.
2. Start the *Sitecore xConnect Search Indexer* Windows service.
3. Start the *Sitecore Marketing Automation Engine* Windows service.
4. Start the *Sitecore Cortex Processing Engine* Windows service.
5. [Rebuild the xConnect indexes.](#)
6. To ensure that all xConnect services are running as expected, review the exceptions in the log files.

### 7.3.2 Exiting Update Mode

After you have configured Sitecore, to complete the upgrade process, you must exit Update Mode in the Update Center. Exiting Update Mode also enables Sitecore xDB.

To exit Update Mode:

1. Log in to the Sitecore LaunchPad and open the **Update Center**.
2. In the **Update Center**, click **Exit update mode**.



Your Sitecore solution is reloaded and xDB along with configuration changes are enabled. The two following topics note potential troubleshooting options after exiting Update Mode.

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

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

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

Ensure you have recompiled your solution with the updated Sitecore DLLs.

## My Sitecore instance is unstable after being updated

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

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

Ensure that you have setup any additional services that are required for your Sitecore XP 9.3.0 instance to run.

### 7.3.3 Using Email Experience Manager (EXM)

#### Important

You can only use EXM when xDB is enabled.

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

EXM is enabled by default.

To enable or disable EXM, in the `\Website\Web.config` file, in the `<appSettings>` section, edit the following setting:

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

The following values are supported:

- yes – default value
- no

### Specify the Email Delivery Provider

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

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

### Enable the Transport Layer Security Protocol

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

For more information about enabling TLS 1.2, see [Microsoft's documentation](#).

### 7.3.4 Redeploy the marketing definitions

To ensure that the binary data stored in the *Data* column in the definition tables - *CampaignActivityDefinitions*, *GoalDefinitions*, and so on – is compatible with this version of Sitecore, you must 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**.

### 7.3.5 Update Sitecore.Analytics.ExcludeRobots.config file

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

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

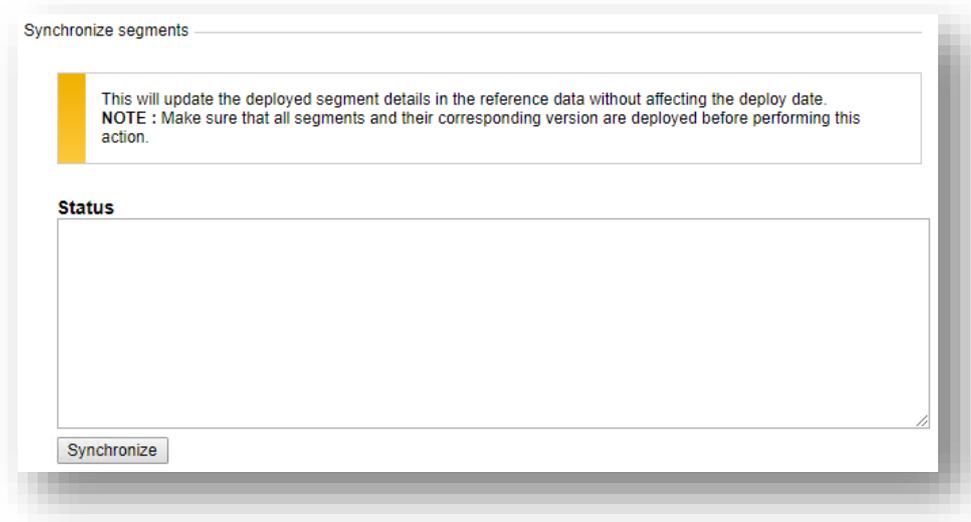
If you have added user agents to the list, you can check whether each user agent is a known robot at <https://51degrees.com/resources/user-agent-tester>. If the user agent is a robot, you can safely remove it from the list.

### 7.3.6 Synchronizing Experience Analytics with Reference Data

In Sitecore XP 9.3.0, in the Experience Analytics *fact\_tables*, the *filterID* column has been added to support system dimensions. *filterID* is not stored in the *Reference Data* database by default.

To store *filterID* in the *Reference Data* database:

1. On the Content Management instance, open the `/sitecore/admin/ExperienceAnalytics.aspx` page.
2. Click **Synchronize**.



#### Important

This step is essential because *Aggregation*, *Reaggregation*, and *Reducer* will not work on segments that are not synchronized properly with the *Reference Data* database.

### 7.3.7 Remove Deprecated Path Analyzer Maps from the Reporting Database

In Sitecore XP 9.3.0, the *First Visits* and *Returning Visits* maps have been removed from the *Global* and the *Site* maps.

After the upgrade, the tree definitions for these maps are still present in the *Reporting* database and Sitecore XP continues to needlessly aggregate data for these definitions.

- To delete the tree definitions of the maps that have been removed, run the following query:

```
DELETE FROM [dbo].[TreeDefinitions]
WHERE DefinitionID IN
('E049AD86-D98B-4639-A450-77D1B294A270', 'D57B53BE-CD98-4EC4-9767-C99F8A72EC0A',
'F2C21FD3-22C5-4F32-92AC-443984D06FC7', '4118EA73-7CEE-4FBC-AECE-5693171744C7')
```

### 7.3.8 Upgrading from Sitecore 9.0.2

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

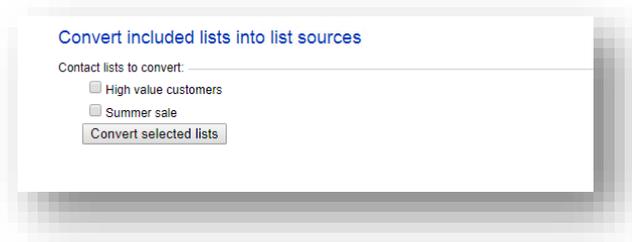
#### Important

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

#### Upgrade the Included Lists in List Manager

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

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



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

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

#### Upgrade the Experience Analytics data

After you upgrade to Sitecore XP 9.3.0, you must upgrade the Experience Analytics data.

To upgrade Experience Analytics data:

- Locate the `SXP_reporting_migrate_experience_analytics_data.sql` script file.

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

#### Important

If you have custom segments, update the migration script and add the affected segments accordingly.

- In SQL Server, select the `Reporting` database and run the `SXP_reporting_migrate_experience_analytics_data.sql` script.

The script deletes the following database artifacts:

- [dbo].Add\_DeviceMetrics\_Tvp
- [dbo].Add\_GeoMetrics\_Tvp
- [dbo].DeviceMetrics\_Type
- [dbo].GeoMetrics\_Type
- [dbo].Fact\_DeviceMetrics
- [dbo].Fact\_GeoMetrics

The script also migrates all the data in the `Fact_DeviceMetrics` and `Fact_GeoMetrics` tables to other tables in the database.

### 7.3.9 Rebuild the Reporting database

When you have completed the upgrade, you must [rebuild the Reporting database](#).

## **7.4 Recompile your solution**

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

Before you deploy your solution in a production environment, we recommend that you test your customizations on a test system.

## Chapter 8

# Sitecore Identity Server

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

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

If you are upgrading from Sitecore 9.1.0 or later, you must upgrade Sitecore Identity Server to version 4.0.0.

This chapter contains the following sections:

- Install Sitecore Identity Server
- Upgrading Sitecore Identity Server

## 8.1 Install Sitecore Identity Server

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

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

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

To install Sitecore Identity Server:

1. [Download](#) the following resources:
  - o Sitecore Identity - Sitecore.IdentityServer.4.0.0-r00257.scwdp.zip
  - o Deployment Configuration files - IdentityServer Deployment Configuration 4.0.0.zip
2. Create a folder named `c:\resourcefiles` and copy both ZIP files to the folder.
3. Unpack the IdentityServer Deployment Configuration 4.0.0.zip file and copy the extracted files to the `c:\resourcefiles` folder.

The folder should now contain the following items:

- o createcert.json
  - o IdentityServer.json
  - o IdentityServer Deployment Configuration 4.0.0.zip
  - o Sitecore.IdentityServer.4.0.0-r00257.scwdp.zip
4. Save your Sitecore license file in the `c:\resourcefiles` folder as `license.xml`.
  5. Create a script file named `install.ps1` and copy the following text into it.

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

#update SIF module
Update-Module SitecoreInstallFramework

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

#install Sitecore Identity Server
$identityParams =
@{
  Path = "$PSScriptRoot\IdentityServer.json"
  Package = "$PSScriptRoot\ Sitecore.IdentityServer.4.0.0-r00257.scwdp.zip"
  SqlDbPrefix = $prefix
  SqlServer = $SqlServer
  SqlSecurityUser = $SqlAdminUser
  SqlSecurityPassword = $SqlAdminPassword
  SitecoreIdentityCert = $certParamsForIdentityServer.CertificateName
  Sitename = $IdentityServer
  PasswordRecoveryUrl = $CMurl
  AllowedCorsOrigins = $CMurl
  ClientSecret = $ClientSecret
```

```

    LicenseFile = "$PSScriptRoot\license.xml"
  }
Install-SitecoreConfiguration @identityParams -Verbose

```

After you copy text from PDFs, it is best practice to review the text for unexpected line breaks.

6. Edit the highlighted sections in the script.
7. Save the script in the `c:\resourcefiles` folder.
8. Open PowerShell as an administrator and navigate to the `c:\resourcefiles` command line and run the following command:

```
.\install.ps1
```

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

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

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

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

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

#### Note

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

## 8.1.1 Configure Sitecore Identity Client

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

To configure a Sitecore instance to use Sitecore Identity Server for logging in, you must configure two parameters:

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

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

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

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

## 8.2 Upgrade Sitecore Identity Server

If you have upgraded from Sitecore 9.1.0 or later, you must upgrade Sitecore Identity Server to version 4.0.0.

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

To upgrade the Sitecore Identity Server:

1. On the Sitecore Identity Server instance, stop the IIS application pool.
2. Extract the content of the `Content\Website\Sitecore.IdentityServer 4.0.0 rev. 00257 (OnPrem)_identityserver.scwdp.zip` file to the existing Sitecore Identity Server directory and replace all the files except:
  - o `Config\production\Sitecore.IdentityServer.Host.xml`
  - o `sitecoreruntime\license.xml`
3. Start the IIS application pool again.

## Chapter 9

# Upgrade Multiple Instances

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

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

In a scaled environment, you must repeat the steps in this guide for each instance, except for the tasks related to the databases, which you must only perform once for each database.

This chapter contains the following section:

- Upgrade a scaled environment with multiple instances

## 9.1 Upgrade a scaled environment with multiple instances

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

### Content Management Servers

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

#### Note

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

### Content Management, Delivery, Processing, and Reporting Servers

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

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

### Scaled XConnect Environment

If you are running the Sitecore XP1 scaled topology, you must perform all the upgrade steps described in the chapter *Upgrade the Sitecore xConnect Server* in alignment with the instructions in the [Sitecore scaling scenarios documentation](#).

## Chapter 10

# Appendix A: The xConnect Application

## Preparation Script

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

To use the script:

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

```

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

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

del App_Data\jobs\continuous\AutomationEngine /q

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

del App_Data\jobs\continuous\IndexWorker /q

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

del App_Data\jobs\continuous\ProcessingEngine /q

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

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

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

```

2. Copy the file to the root folder of the xConnect application.
3. In a command line, run the `.bat` file in the xConnect root folder.

## Chapter 11

# Appendix B: Generating shard keys and missed percentile values

This appendix applies if the *Collection* database is on SQL Server and the *AlwaysEncrypted* feature is not configured. For more information, see the section *xDB Collection database on SQL Server*.

The *Sitecore Xdb Collection Update* tool is used to generate shard keys and missed percentile values from record IDs. These values are calculated from row IDs (usually from contact ID).

We recommend that you replace the default values of the shard key and the percentile with the value that is calculated based on row ID.

When you update the SQL *Collection* schema to version 9.3.0, the shard key values are null by default. Previous upgrades to version 9.2 might result in percentile value being less than 0.

## 11.1 Run the Sitecore Xdb Collection Update tool

In Sitecore XP 9.3.0, the Collection database schema has been changed to store a shard key value. This is required by the Microsoft Split-Merge service that rebalances data between shards.

This functionality is only available if you are running Sitecore XP 9.3.0 on Azure (not on-Prem), and do not use the *Always encrypted* feature. It is only relevant if you want to split the data in databases to improve scaling.

The *Sitecore Xdb Collection Update* tool fills in the shard key values for data that has been collected before you upgraded to Sitecore XP 9.3.0. You can use it to update production databases while only placing a minimal load on the database server.

Download the *Sitecore Xdb Collection Update* tool from the [Sitecore Experience Platform download site](#).

The tool is stored in the *Configuration files for upgrade* file in the `\Tools\XConnect Upgrade Tool` folder. You might have downloaded the *Configuration files for upgrade* file earlier.

### Important

You must run the *Sitecore Xdb Collection Update* tool before you can use the Microsoft Split-Merge service.

For more information about how to configure Sitecore XP to use the Microsoft Split-Merge service, see the [Sitecore documentation](#).

### 11.1.1 Run the tool

You run the *Sitecore Xdb Collection Update* tool from a Windows Command Prompt.

You can also run this tool before using the shard keys *functionality*.

### Note

We recommend that you run this tool on offline databases.

If it is impossible to take the database offline, we recommend that you [turn off change tracking](#) for the tables whose shards are updated - *Contacts*, *ContactFacets*, *Interactions*, *InteractionFacets*.

### Important

Do not turn off change tracking at database level.

### 11.1.2 Update operations

The *Sitecore Xdb Collection Update* tool initiates several update operations. The update operations use the primary key to scan the relevant tables by ID in ascending order, if possible. Next operations are also supported.

The update operations are:

- *deviceProfiles* – Updates the *shard key* column in the *device profile* table. The value is calculated based on the device profile ID.
- *deviceProfileFacets* – Updates the *shard key* column in the *device profile facets* table. The value is calculated based on the device profile ID.
- *contacts* – Updates the *shard key* column in the *contact*, *contact's identifiers*, *contact's facets*, *contact's interactions*, and *interaction's facets* tables. The shard key value is calculated

based on the contact ID for all these tables and should be the same for all contact-related data.

- *contactIdentifierIndexes* - Updates the *shard key* column for the *contact identifier index* table. The value is calculated based on the contact ID.
- *contactPercentile* - Updates the *percentile* column of the *contact* table. The value is calculated based on the contact ID.
- *interactionPercentile* - Updates the *percentile* column of the *interaction* table. The value is calculated based on the interaction ID.

The update operations are listed in the recommended run order.

#### Note

If you turned off change tracking on shard tables, turn it on again when the upgrade tool is finished.

### 11.1.3 Tool arguments:

The following table lists the Sitecore *Xdb Collection Update* tool's arguments:

Argument (short synonym)	Required	Default value	Description
--cmd (-c)	Yes		The update operation that you want to run.
--connectionString(-s)	Yes		The connection string to the collection shard, including the user ID and password.
--consumersNumber (-cn)	No	8	The number of threads that will send update queries.
--batchSize (-bs)	No	500	The number of entities that should be updated in a single query.
--timeout(-t)	No	30	The command query timeout.
--startId (-si)	No	00000000-0000-0000-0000-000000000000	The ID value that you want to start updating from. The <i>contactPercentile</i> , <i>contactPercentile</i> , and <i>contacts</i> update operations support this argument.

Example:

```
.\SitecoreXdbCollectionUpdateTool.exe -s "user id=sa;password=12345;data source=.\SQL2017;Initial Catalog=xPFPlatformTests_Xdb.Collection.Shard0" -c contacts
```