Sitecore Experience Platform Upgrade Guide

Sitecore Experience Platform 10.0.0

Upgrade to Sitecore XP 10.0.0 from Sitecore XP 9.0.2 or later

Table of Contents

Chapter 1

Introduction

This guide describes how to upgrade from Sitecore Experience Platform 9.0.2 or later to Sitecore Experience Platform 10.0.0 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 10.0.0 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

Prepare for the upgrade

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

Before you upgrade to Sitecore XP 10.0.0, 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 Installation Guide.

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

This chapter contains the following sections:

- Requirements

- Prerequisites for upgrading Sitecore XP

- Prerequisites for upgrading xConnect

- Backup and disable customizations

- Disable the Sitecore modules

- Update the connection strings

- Stop analytics and the xConnect services

- The Sitecore Update Center

- Entering update mode

2.1 Requirements

Before you upgrade to Sitecore XP 10.0.0, you should ensure that you meet the software and hardware requirements for this version of Sitecore XP. You must also download the upgrade files and prepare your environment for the upgrade process.

For more information about the requirements, see the Sitecore XP 10.0.0 Installation Guide.

Note
Since Sitecore XP 9.3.0, you can no longer use Lucene as your search provider.

2.1.1   Obtain and set up certificates for SSL

Note
If you already use HTTPS, skip this section.

Since 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 Prerequisites for upgrading Sitecore XP

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 every version of Sitecore XP, you also require the following file which is listed on the download site as *Configuration files for upgrade*:

- `Sitecore 10.0.0 rev. 004346 (upgrade files).zip`

  This file contains several files including:

  o `Database Upgrade Script.zip`

    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.

  o `MarketingDefinitionsUpgrade.zip`

    This file contains the `.aspx` pages that you use to prepare your marketing definitions for the upgrade.

2.3 Prerequisites for upgrading xConnect

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

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 10.0.0.

Before you upgrade xConnect:

1. Download the following files:

   o `Sitecore 10.0.0 rev. 004346 (WDP XP0 packages).zip`.

      In the *Download options for On Premises deployment* section, this file is called *Packages for XP Single*.

   o `Sitecore 10.0.0 rev. 004346 (upgrade files).zip`.

      In the *Upgrade options* section, this file is called *Configuration files for upgrade*.

2. The `Sitecore 10.0.0 rev. 004346 (WDP XP0 packages).zip` file contains the `Sitecore 10.0.0 rev. 004346 (OnPrem)_xp0xconnect.scwdp.zip` file.

3. Unpack the `Sitecore 10.0.0 rev. 004346 (upgrade files).zip` file and then unpack the `Database Upgrade Script.zip` file.

   This file contains several database upgrade scripts.

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.

2.4 Backup and disable customizations

Sitecore XP

The new version of Sitecore XP might contain some breaking changes. To separate the upgrade of Sitecore XP from the upgrade of any customizations that you have made, move your custom DLLs 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 then move the files back.

xConnect

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 and configuration file patches.

- Custom xConnect model JSON files.

- Custom DLL files.

You must also ensure that the default Sitecore XP and xConnect 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.

2.5 Disable the Sitecore modules

You must disable any Sitecore modules that you are using.

After you upgrade Sitecore XP, you must upgrade all the individual modules.

2.5.1   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, add `.disabled` to the file extension of the following configuration files:

- o `\App_Config\Include\Sitecore.WFFM.Speak.config`
- o `\App_Config\Include\Sitecore.WFFM.Services.config`
- o `\App_Config\Include\Sitecore.WFFM.Dependencies.config`
- o `\App_Config\Include\Sitecore.WFFM.Analytics.config`
- o `\App_Config\Include\Sitecore.Forms.config`
- o `\App_Config\Include\Sitecore.MvcForms.config`

2.5.2   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.

2.5.3   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.

2.6 Update the connection strings

Some connections strings must be updated before you upgrade to Sitecore XP 10.0.0. To do this open the `\App_Config\ConnectionStrings.config` file and follow the instructions in this section.

2.6.1 Add the Security Connection String

Important
This section applies only if you are upgrading from Sitecore 9.0.2.

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

- Add the *security* connection string, for example:

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

If you do not plan to move security to another database, you must still add the *security* connection string and ensure that it points to the *Core* database.

2.6.2 Remove Xdb.MarketingAutomation Connection String on CM

In Sitecore XP 10.0.0, the `xdb.marketingautomation` connection string has been removed and is no longer required on CM instances.

To delete the `xdb.marketingautomation` connection string:

- In the `sitecore\App_Config\ConnectionStrings.config` file, delete:

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

2.7 Stop analytics and the xConnect services

If you are using the xDB functionality, you must complete the steps described in this section.

2.7.1 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.

2.7.2 Disable Experience Analytics

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

1. In the `Sitecore 10.0.0 rev. 004346 (upgrade files).zip` file, locate the `Disable xDB.zip` file and unpack it to find the `Disable.xAnalytics.config` file.

2. Copy the `Disable.xAnalytics.config` file to the `\App_Config\Update\` folder.

Important:
The `Update` folder is only visible when you enter update mode. Before you exit update mode, remove the `Disable.xAnalytics.config` file.

2.7.3 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:

    o Stop scheduling new processing tasks.

    o Wait for all the ongoing and pending tasks to complete.

    o 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:

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

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

2.8 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.

The Update Center accesses a package management service for information about upgrade and for the upgrade packages themselves. Finally, the Update Center can install updates from a package in the file system.

You open Update Center from the Sitecore Dashboard:
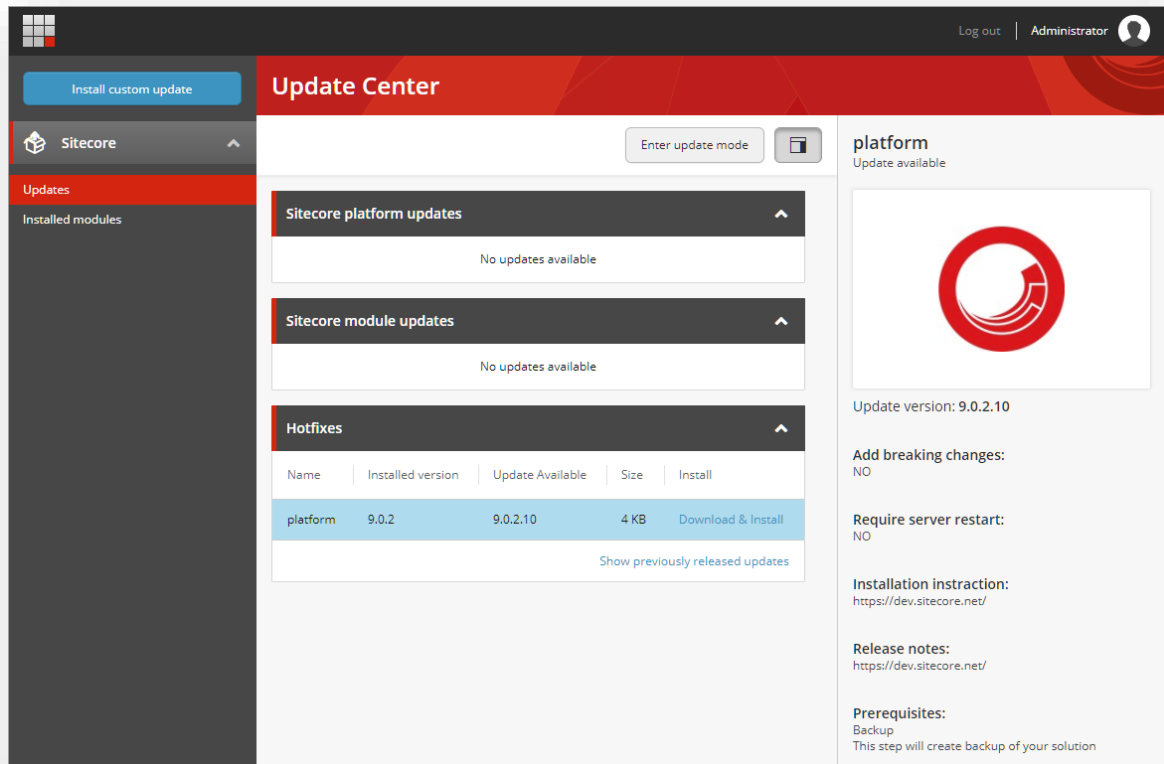
The left pane, contains three options:

- **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 notes 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.

2.8.1   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`.

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.

2.8.2   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/packagemanagementservice/" />
```

2.9 Entering update mode

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:

    o 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.

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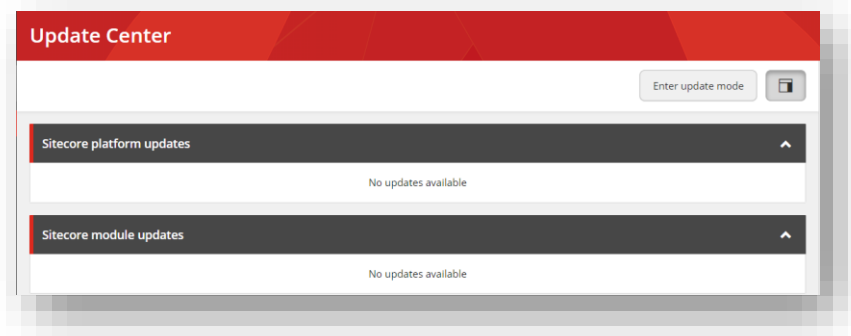
> **Note**
> Ensure that Internet Information Services (IIS) is configured to allow access to the `\sitecore\admin` folder.

Chapter 3

Upgrade the databases

Depending on which version of Sitecore you are updating from you must upgrade some existing databases and deploy some others.

- The databases

3.1 The databases

When you upgrade to Sitecore XP 10.0.0 from different versions of Sitecore XP, you must upgrade or deploy all the appropriate databases.

Database upgrade

The database upgrade scripts contain the changes that must be applied to the Sitecore databases 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.

Before you upgrade a database:

1. Extract the database upgrade scripts from the `Database Upgrade Script.zip` file that you downloaded earlier.

2. Read the instructions for the database that you want to upgrade.

   You may need to run the scripts in a certain sequence, or to perform some manual actions.

Database deployment

If you are upgrading from Sitecore XP 9.0.2, you must deploy the *processing_engine_storage* and *processing_engine_tasks* databases.

For more information about deploying these databases, see the section *Install and configure Sitecore Cortex Processing Engine*.

The following table lists all the databases that you need to upgrade or deploy:

Type	DB name	Sitecore version				
		9.0.2	9.1.0	9.1.1	9.2.0	9.3.0
XM	core	Upgrade	Upgrade	Upgrade	Upgrade	Upgrade
	master	Upgrade	Upgrade	Upgrade	Upgrade	Upgrade
	web	Upgrade	Upgrade	Upgrade	Upgrade	No action
	experienceforms	Upgrade	Upgrade	Upgrade	Upgrade	Upgrade
xDB	exm_master	Upgrade	Upgrade	No action	No action	No action
	reporting	Upgrade	Upgrade	Upgrade	Upgrade	Upgrade
	processing_tasks	Upgrade	Upgrade	Upgrade	Upgrade	No action
	collection_shards	Upgrade	Upgrade	Upgrade	Upgrade	Upgrade
	collection_smm	Upgrade	Upgrade	Upgrade	Upgrade	No action
	referencedata	Upgrade	Upgrade	Upgrade	Upgrade	No action
	marketingautomation	Upgrade	Upgrade	Upgrade	Upgrade	Upgrade
	messaging	Upgrade	Upgrade	Upgrade	Upgrade	No action
	processing_pools	Upgrade	Upgrade	Upgrade	Upgrade	No action

Type	DB name	Sitecore version				
		9.0.2	9.1.0	9.1.1	9.2.0	9.3.0
	processing_engine_storage	Deploy	Upgrade	Upgrade	Upgrade	No action
	processing_engine_tasks	Deploy	Upgrade	Upgrade	Upgrade	No action

3.1.1   The XM databases

To upgrade the XM databases:

Core

1. If you are upgrading from Sitecore XP 9.2.0 or earlier, run the `CMS_core_master_web9x.sql` script.

2. Run the `CMS_core.sql` script.

Master

1. If you are upgrading from Sitecore XP 9.2.0 or earlier, run the `CMS_core_master_web9x.sql` script.

2. Run the `CMS_master.sql` script.

Web

- Run the `CMS_core_master_web9x.sql` script.

ExperienceForms

1. Run the `SXP_experienceforms_storage.sql` script.

2. If you are upgrading from Sitecore XP 9.2.0 or earlier, run the `SXP_experienceforms_filestorage.sql` script.

3.1.2   The xDB databases

EXM.Master

1. If you are upgrading from Sitecore XP 9.0.2, run the `SXP_exm_master_dispatchqueue.sql` script for the *EXM.Master* database.

2. If you are upgrading from Sitecore XP 9.0.2 - 9.1.0, run the following scripts for the *EXM.Master* database:
   - `SXP_exm_master_suppressions.sql`
   - `SXP_exm_master_save_suppressions.sql`
   - `SXP_exm_master_update_suppressions.sql`

Reporting

To upgrade the *reporting* database:

1. Run the `SXP_reporting.sql` script for the *Reporting* database.

   If you are upgrading from Sitecore XP 9.2.0 or earlier, one of the things that this script does is add two new columns – *Converted* and *FirstImpressionCount* – to some of the

reporting database artifacts. These columns are used to improve the accuracy of conversion and bounce rate calculations.

The script sets the default value of the columns to 0. This means that conversion and bounce rate of old data is set to zero and will be calculated for new interactions after the upgrade.

Note:
Your historical data will not be changed to zero if you are upgrading from 9.3.0 or later.

2. If you are upgrading from Sitecore XP 9.2.0 or earlier, 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.

Processing_tasks

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

Collection - on SQL Server and SQL Azure

Important
If the c*ollection* database is on SQL Server, follow the procedures described in this section. Otherwise, skip this section.

To upgrade the *Shard Map Manager* database:

- Run the `SXP_collection_smm.sql` script for the *Shard Map Manager* database.

  This script registers the schemas for the shard maps.

To upgrade the shards:

- If the *Always Encrypted* feature *is* configured for the *collection* database:

  o Run the `SXP_collection_AE_Part1.sql` script for every shard.

  Or

- If the *Always Encrypted* feature *is not* configured for the *collection* database:

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

- If you are upgrading from Sitecore XP 9.0.2:

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

  You must run this script whether or not the *Always Encrypted* feature *is* configured for the *xDB Collection* database.

  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]
```

- o If the *Always Encrypted* feature *is* configured for the *collection* database 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](#).

If your *collection* database is deployed on SQL Azure and you are going to use the split/merge service to distribute analytics data and your collection databases, follow the procedures described here. Otherwise skip next steps.

- o Run the *Sitecore Xdb Collection Update* tool to generate shard keys and any missing percentile values from the record IDs.

    For more information, see **Error! Reference source not found.**.

- o After you have run the *Sitecore Xdb Collection Update* tool, if the *Always Encrypted* feature *is* configured for the *collection* database run the `SXP_collection_AE_Part2.sql` script for every shard.

    Otherwise run the `SXP_collection_Part2.sql` script for every shard.

Collection database on MongoDB

Important
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 10.0.0 describes how to configure the MongoDB provider for xConnect.

To continue using the MongoDB xDB Collection database provider with Sitecore XP 10.0.0, you must upgrade your existing MongoDB instance to version 4.0.5 – 4.0.13.

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](#).

referencedata

We recommended that you make a copy of the r*eferencedata* database before completing this procedure.

Important
If you are upgrading a Sitecore XP instance that is deployed in Azure App Service, you must change the collation of the *referencedata* database before you upgrade it. You must [change](#) the collation from *SQL_Latin1_General_CP1_CI_AS* to *Latin1_General_CS_AS*.

To upgrade the *referencedata* database:

1. Run the `SXP_referencedata_Part1.sql` script.

2. 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.

3. Run the `SXP_referencedata_Part2.sql` script.

marketingautomation

To upgrade the *marketingautomation* database:

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

messaging

To upgrade the *messaging* database:

- Run the `SXP_Messaging.sql` script for the *Messaging* database.

processing_pools

To upgrade the *processing_pools* database:

- Run the `SXP_processing_pools.sql` script for the *processing_pools* database.

processing_engine_storage

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 privileged user that exists in the *processing_engine_storage* database.

   This updates the permissions for the least privileged user in the *processing_engine_storage* database.

processing_engine_tasks

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 privileged user that exists in the *processing_engine_tasks* database.

   This updates the permissions for the least privileged user in the *processing_engine_tasks* database.

Chapter 4

Upgrade the 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:

o Sitecore XP 9.0.2

o Sitecore XP 9.1.0 (Initial Release that introduced the Sitecore Cortex™ Processing Engine role)

o Sitecore XP 9.1.1

o Sitecore XP 9.2.0

o Sitecore XP 9.3.0

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:

- Update the files

- Configure xConnect server

- Install and configure Sitecore Cortex Processing Engine

4.1 Update the files

4.1.1   Copy the new files

Copy the new files to 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: Scripts*

The xConnect Application Preparation Script.

To update the files:

1. Delete all the files under the following folders, but leave the folders themselves to ensure that the file permissions are preserved:
   - `\Bin`
   - `\App_Data\collectiondeployment`
   - `\App_Data\Config`
   - `\App_Data\Copyrights`
   - `\App_Data\Diagnostics`
   - `\App_Data\Models`
   - `\App_Data\solrcommands`
   - `\App_Data\jobs\continuous\AutomationEngine`
   - `\App_Data\jobs\continuous\AutomationEngine\App_Data\Config`
   - `\App_Data\jobs\continuous\AutomationEngine\App_Data\Diagnostics`
   - `\App_Data\jobs\continuous\IndexWorker`
   - `\App_Data\jobs\continuous\IndexWorker\Copyrights`
   - `\App_Data\jobs\continuous\IndexWorker\App_Data\Config`
   - `\App_Data\jobs\continuous\IndexWorker\App_Data\Diagnostics`
   - `\App_Data\jobs\continuous\IndexWorker\App_Data\Models`
   - `\App_Data\jobs\continuous\ProcessingEngine`
   - `\App_Data\jobs\continuous\ProcessingEngine\App_Data\Config`
   - `\App_Data\jobs\continuous\ProcessingEngine\App_Data\Diagnostics`
   - `\App_Data\jobs\continuous\ProcessingEngine\App_Data\Models`

   Note
   If you are upgrading from SC XP 9.0.2, you do not have the `\App_Data\jobs\continuous\ProcessingEngine` folder.

2. In the `\App_Data\jobs\continuous\IndexWorker` folder, delete the language folders like `\de-DE` if they exist.

   A complete list of the language folders that you should delete is available in ***Error! Reference source not found.*.**

3. Copy all the files and folders from the `Sitecore 10.0.0 rev. 004346 (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.

4.1.2   Manually update the configuration files

You must manually update a number of configuration files.

- In 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`

   Take appropriate files from the `Sitecore 10.0.0 rev. 004346 (OnPrem)_xp0xconnect.scwdp\Content\Website` folder that you unpacked earlier and apply all the customization that you need.

4.1.3   Add the *xconnect.search* connection strings

In Sitecore XP 10.0.0, there are separate connection strings for the *xconnect.collection* and *xconnect.search* services for the CM and Marketing Automation roles.

To add the `xconnect.search` and `xconnect.search.certificate` connection strings to your CM and Marketing Automation instances:

1. In the `sitecore\App_Config\ConnectionStrings.config` file, add:

```
<add name="xconnect.search" connectionString="https://Sitecore_search" />
<add name="xconnect.search.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;Fi
ndValue=search_certificate_thumbprint" />
```

2. In the `marketingautomation\App_Config\ConnectionStrings.config` file, add:

```
<add name="xconnect.search" connectionString="https://Sitecore_search" />
<add name="xconnect.search.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;Fi
ndValue=search_certificate_thumbprint" />
```

3. Update the existing *xconnect.collection* connection strings:

```
<add name="xconnect.collection" connectionString="https://Sitecore_collection" />
<add name="xconnect.collection.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;Fi
ndValue=collection_certificate_thumbprint" />
```

4.1.4   Add the *reporting* connection strings

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

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

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

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

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

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

4.2 Configure xConnect server

4.2.1 Reconfigure the 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"
```

4.2.2 Reconfigure the 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"
```

4.2.3 Reconfigure the SQL Provider for the Collection database

Important
If you are using the *Always Encrypted* feature, you must reconfigure the SQL Provider for the *Collection* database. Otherwise, skip this section.

In Sitecore XP 10.0.0, 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.IContactsExecutionS
trategy>
      <Type>Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.ContactsStagin
gTableExecutionStrategy, Sitecore.Xdb.Collection.Data.SqlServer</Type>
      <As>Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.IContactsExecuti
onStrategy, Sitecore.Xdb.Collection.Data.SqlServer</As>
      <LifeTime>Singleton</LifeTime>
</Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.IContactsExecutionStrategy
>
<Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.IInteractionsExecutionStrat
egy>
      <Type>Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.InteractionsSt
agingTableExecutionStrategy, Sitecore.Xdb.Collection.Data.SqlServer</Type>
      <As>Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.IInteractionsExe
cutionStrategy, Sitecore.Xdb.Collection.Data.SqlServer</As>
      <LifeTime>Singleton</LifeTime>
</Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.IInteractionsExecutionStra
tegy>
```

```
<Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.IDeviceProfilesExecutionStr
ategy>
      <Type>Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.DeviceProfiles
StagingTableExecutionStrategy, Sitecore.Xdb.Collection.Data.SqlServer</Type>
      <As>Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.IDeviceProfilesE
xecutionStrategy, Sitecore.Xdb.Collection.Data.SqlServer</As>
      <LifeTime>Singleton</LifeTime>
</Sitecore.Xdb.Collection.Data.SqlServer.Managers.ExecutionStrategy.IDeviceProfilesExecutionSt
rategy>
```

4.2.4 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.

4.3 Install and configure Sitecore Cortex Processing Engine

Important

If you are upgrading from Sitecore XP 9.0.2, you must install and configure the Sitecore Cortex Processing Engine. For more information about installing Sitecore xConnect, see the Sitecore Experience Platform 10.0.0 Installation Guide.

If you are upgrading from Sitecore XP 9.1.0 or later with xConnect installed, the Sitecore Cortex Processing Engine is already installed and you can skip this chapter.

When you updated the files, you installed the files for the Cortex Processing Engine. Now you must deploy the Processing Engine databases and configure the Processing Engine service.

When you deploy a new database, you must add a connection string to the appropriate Sitecore role. To restrict access to the database, you must create a user with limited access rights and specify this user in the connection string. If you don't have a least privileged user in the existing databases, you must create one. You can see an example of a script that creates a least privileged user in *Appendix A: Scripts*.

The WDP packages contain sample scripts for creating users for each database.

To deploy the databases, in the `Sitecore 10.0.0 rev. 004346 (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, 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.

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

4. 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.

5. 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
tecore.ProcessingEngine.exe -service" start= auto obj= "NT AUTHORITY\LOCAL SERVICE"
DisplayName= "Sitecore Cortex Processing Engine - instanceName.xconnect-
ProcessingEngineService"
```

6. 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" />
```

4.3.1   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 10.0.0 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 10.0.0, you must have completed all the prerequisite steps that are described in this guide.

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

This chapter contains the following sections:

- Install the Upgrade Package

- Troubleshooting

- Review changes and update configuration files

5.1 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 the chapter *Prepare for the* upgrade.

To install the upgrade package:

1. On the Sitecore Launchpad, open the **Update Center**.
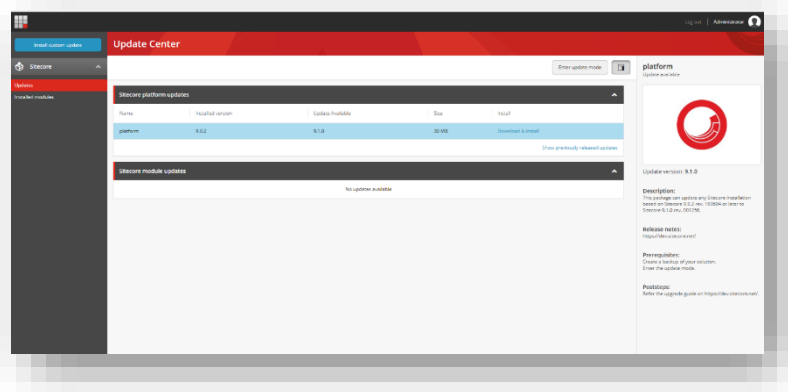
2. Select the Sitecore XP 10.0.0 upgrade package and then click **Download and Install**.

3. When the package is downloaded, in the **Ready to install** dialog box, click **Install**:
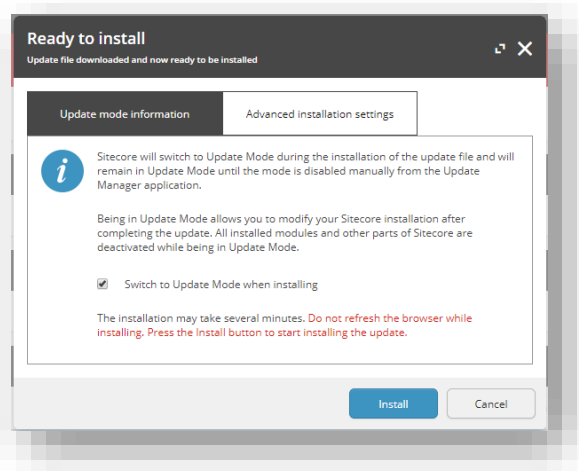
4. Wait until the package is installed.

Sometimes, the final part of the upgrade process fails, and you see the *Installation failed!* message.

If this happens:

1. Select the Sitecore XP 10.0.0 upgrade package and then click Download and Install

2. When the package is downloaded, in the **Ready to install** dialog box, click the **Advanced installation settings** tab and clear all the options before *Execute post-steps*.

3. Click **Install** and then wait until the package is installed.

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

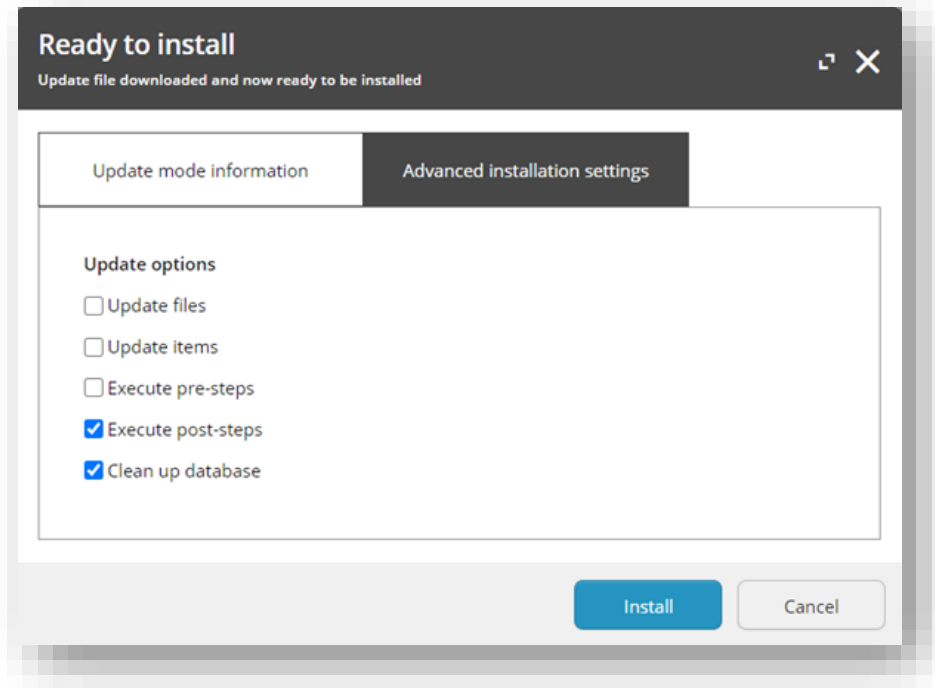## 5.2 Troubleshooting

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

I cannot see an update to Sitecore XP 10.0.0 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.

5.3 Review changes and update configuration files

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

5.3.1   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.

5.3.2   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.

In Sitecore XP 10.0.0, the Experience Analytics configuration files have been updated.

If you have enabled reaggregation before the upgrade, you must delete the old reaggregation configuration files.

The reducer configuration files are now disabled by default and you must ensure that these configuration files are not present in `App_Config\Sitecore\ExperienceAnalytics` folder.

New versions of the reaggregation and reducer configuration files are stored in the `App_Config\Include\Examples` folder. You must ensure that they are disabled until the whole upgrade process is finished.

5.3.3   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.

5.3.4   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.

5.3.5   Search provider settings

Since Sitecore XP 9.0, the search provider is specified in the `Web.config` file. After the upgrade, 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:

  o Solr – default value since Sitecore XP 9.2.0.

  o 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/100/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/100/platform-administration-and-architecture/en/configure-azure-search-for-xconnect.html

5.3.6   Server role settings

After you upgrade to Sitecore XP 10.0.0, 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 upgrade 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:

- o ContentDelivery

- o ContentManagement

- o ContentDelivery, Indexing (introduced in Sitecore XP 9.2.0)

- o ContentManagement, Indexing (introduced in Sitecore XP 9.2.0)

- o Processing

- o Reporting

- o 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:

  - o `Solr` – the default value is Solr since Sitecore XP 9.2.0.

  - o `Azure` – Azure Search

Note
Support for the Lucene search provider was removed in Sitecore XP 9.3.0.

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

Chapter 6

Upgrading Solr

To continue using the Solr or SolrCloud search provider in Sitecore XP 10.0.0, you must upgrade your Solr environment to version 8.4.0.

This chapter contains the following sections:

- Upgrade Solr

6.1 Upgrade Solr

Important
During the upgrade process, you must keep Sitecore XP configured to use the previous Solr environment. After you have upgraded Sitecore XP and xConnect, you can reconfigure them to use the new Solr environment.

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_4/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. You must recreate them and then rebuild the indexes.

Important
Sitecore XP 9.3.0 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/100/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/100/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/100/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/100/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

- Resume the xConnect services

- Exiting update mode

- 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:

- Apply an important hotfix.

  After you install Sitecore XP 10.0.0, you *must* install a hotfix before you publish your website.

  You can download the hotfix and read more about it on the [Sitecore Knowledge Base](#).

- 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:

  o Danish (da-DK)

  o German (de-DE)

  o Japanese (ja-JP)

  o 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, use the Redis Cache service Console to clear the cache.

2. Start all the Sitecore XP instances.

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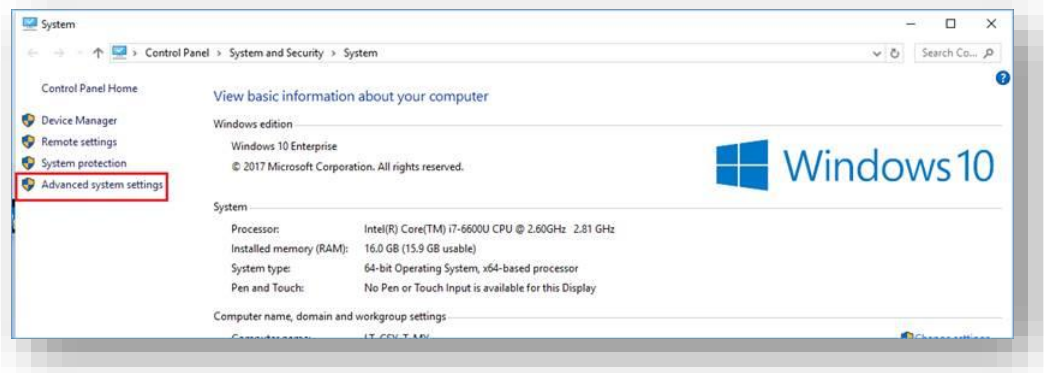
2. In the **System Properties** dialog, click **Environment Variables**.
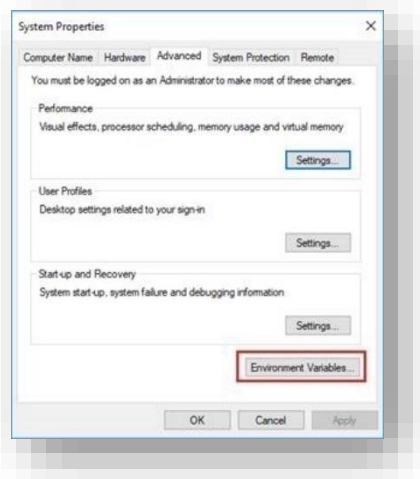
3. In the **Environment Variables** dialog, in the **System variables** section, click **New**.

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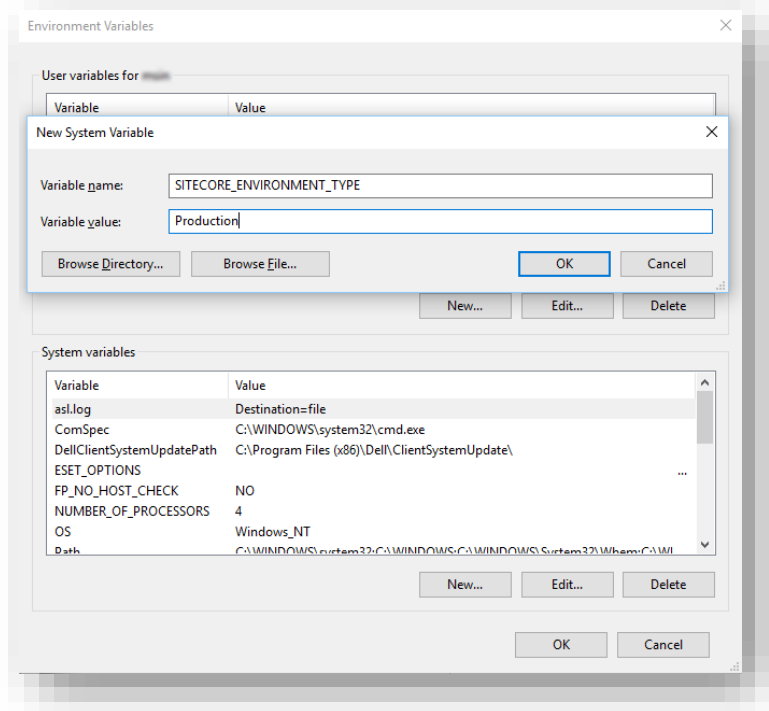

5. Restart the computer.

7.3 Resume the xConnect services

Note
If you do not use xConnect, skip this chapter.

You must restart the xConnect services that you stopped in the chapter *Upgrade the databases*.

You must restart these services before you enable xDB or exit update mode in the Update Center.

To restart 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.

7.4 Exiting update mode

After you have configured Sitecore, to complete the upgrade process, you must exit Update Mode in the Update Center. When you exit update mode, Sitecore xDB is enabled.

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.**
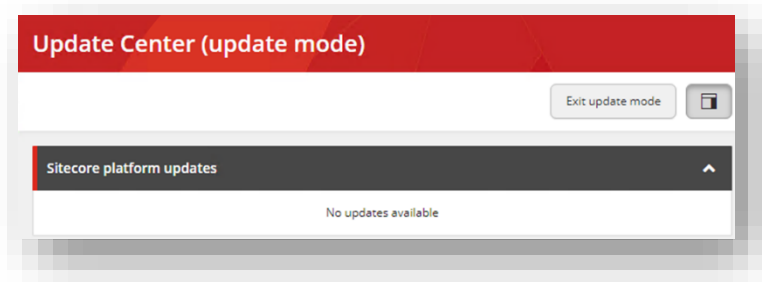
Your Sitecore solution is reloaded and xDB along with configuration changes are enabled.

The two following topics note potential troubleshooting options after you exit 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 10.0.0 instance to run.

7.5 xDB Maintenance

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

7.5.1   Using Email Experience Manager (EXM)

Important
To use EXM, you must have an EXM license.

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.

7.5.2   Redeploy the marketing definitions

To ensure that the binary data stored in the *Data* column in the *CampaignActivityDefinitions*, *GoalDefinitions* tables*,* 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**.

7.5.3 Update the Sitecore.Analytics.ExcludeRobots.config file

In the
`website\App_Config\Sitecore\Marketing.Tracking\Sitecore.Analytics.ExcludeRo
bots.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.

7.5.4 Deactivate unused segment definitions in the Reference Data database

In Sitecore XP 10.0.0, if you delete a segment definition, it is automatically deactivated in the *Reference Data* database.

If you are using the Sitecore Experience Database (xDB) functionality and have deleted any segment definitions from your content management server in an earlier version of Sitecore XP, you must deactivate these segment definitions in the *Reference Data* database.

To deactivate the unused segment definitions:

1. Unpack the `MarketingDefinitionsUpgrade.zip` file.

2. Copy the `DeactivateSegmentsInReferenceData.aspx` page to the `\sitecore\admin` folder of your website.

3. To open the DeactivateSegmentsInReferenceData.aspx page, enter the following URL in a browser:

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

4. Click **Test Run - Get affected segments**.

   The segment definitions are validated.

   If any segment definitions are not found under the `/sitecore/system/Marketing Control Panel/Segments` item on the Content Management server, a list of these segment definitions that should be deactivated in the *Reference Data* database is shown.

5. Click **Deactivate segments**.

   The segment definitions are deactivated in the *Reference Data* database.

6. To validate that all the segment definitions were successfully deactivated, click **Test Run - Get affected segments**.

7.5.5 Synchronizing Experience Analytics Segments

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.

In Sitecore XP 10.0.0, segment details are distributed between the *Reference Data* and *Reporting* databases. You must synchronize the segments across these databases to ensure that *Aggregation*, *Reaggregation*, and *Reducer* still work on segments.

To synchronize the segment details across the *Reference Data* and *Reporting* databases:

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

2. Click **Synchronize**.
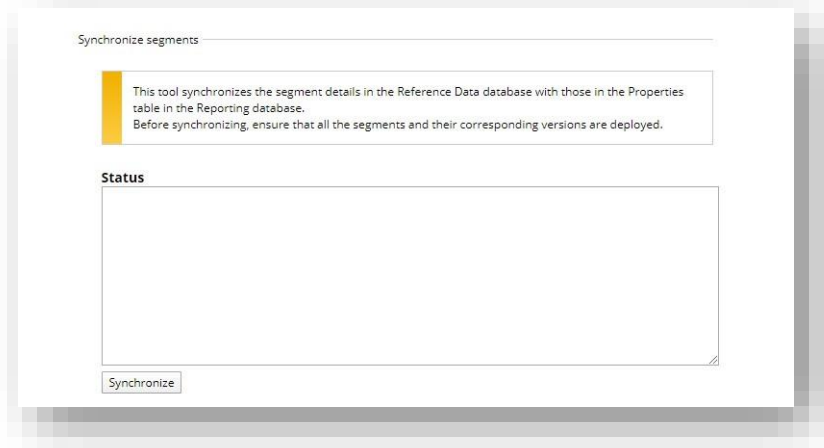
If you are upgrading from Sitecore 9.2 or earlier, and you have created custom segments that uses filters, you must update the system dimension.

To update the system dimension:

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

2. Click **Update System Dimension**.
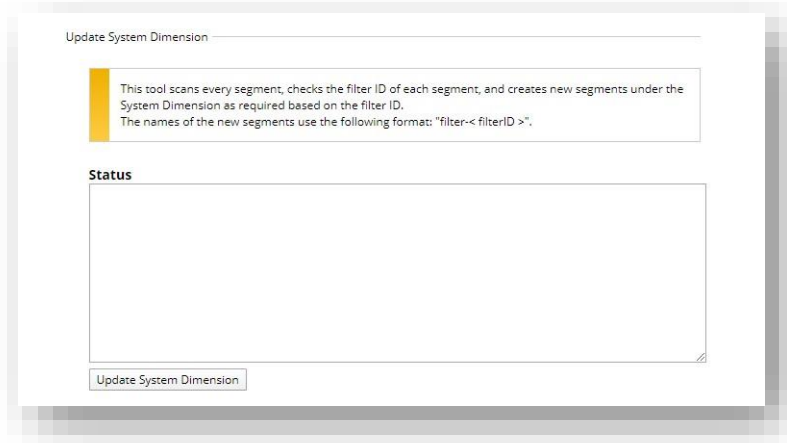
7.5.6 Remove deprecated Path Analyzer maps from the Reporting database

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')
```

7.5.7   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:
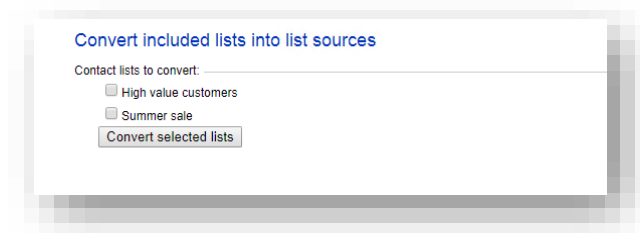
1. On the Content Management instance, open the
    `/sitecore/admin/ConvertContactListSources.aspx` page.

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

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

Upgrade the Experience Analytics data

After you upgrade to Sitecore XP 10.0.0, you must upgrade the Experience Analytics data.

To upgrade Experience Analytics data:

1. Locate the `SXP_reporting_migrate_experience_analytics_data.sql` script file.

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

    Important
    If you have added custom segments, you must update the migration script with these segments.
    For an example, see the appendix *Update Experience Analytics custom segments*.

2. 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:

- o `[dbo].Add_DeviceMetrics_Tvp`
- o `[dbo].Add_GeoMetrics_Tvp`
- o `[dbo].DeviceMetrics_Type`
- o `[dbo].GeoMetrics_Type`
- o `[dbo].Fact_DeviceMetrics`
- o `[dbo].Fact_GeoMetrics`

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

7.5.8   Rebuild the Reporting database

If you want to ensure that your historical and current data is consistent in the *Reporting* database, you must rebuild the Reporting database.

For more information about why you should rebuild the reporting database, see this topic.

7.6 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 5.0.0.

This chapter contains the following sections:

- Install Sitecore Identity Server

- Upgrade Sitecore Identity Server

8.1 Install Sitecore Identity Server

If you are upgrading 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 2.1.18 Runtime or later.

To install Sitecore Identity Server:

1. Download the following resources:
   - `Sitecore.IdentityServer.5.0.0-r00277.scwdp.zip`
   - `IdentityServer Deployment Configuration 5.0.0.zip`

2. Create a folder named `c:\resourcefiles` and copy both ZIP files to the folder.

3. Unpack the `IdentityServer Deployment Configuration 5.0.0.zip` file and copy the extracted files to the `c:\resourcefiles` folder.

   The folder should now contain the following items:
   - `createcert.json`
   - `IdentityServer.json`
   - `Install-IdentityServer.ps1`
   - `IdentityServer Deployment Configuration 5.0.0.zip`
   - `Sitecore.IdentityServer.5.0.0-r00277.scwdp.zip`

4. Save your Sitecore license file in the `c:\resourcefiles` folder as `license.xml`.

5. Edit the `Install-IdentityServer.ps1` file, update the required parameters, and save the changes.

6. Open PowerShell as an administrator and go to the `c:\resourcefiles` command line and run the following cmdlet:

   ```
   .\Install-IdentityServer.ps1
   ```

7. 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.

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

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

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

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 are upgrading from Sitecore 9.1.0 or later, you must upgrade Sitecore Identity Server to version 5.0.0.

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 5.0.0 rev. 00277 (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 databases* in alignment with the instructions in the Sitecore scaling scenarios documentation.

Chapter 10

Appendix A: Scripts

The xConnect Application Preparation Script

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

To use the script:

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

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

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

del App_Data\jobs\continuous\AutomationEngine /q

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

del App_Data\jobs\continuous\IndexWorker /q

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

del App_Data\jobs\continuous\ProcessingEngine /q

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

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

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

2. Copy the file to the root folder of the xConnect application.

3. In a command line, run the `.bat` file in the xConnect root folder.

User credential script for database deployment

The following script is an example of how to specify user credentials when you deploy a database.

You must update the script to match the database.

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

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

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

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

Update Experience Analytics custom segments

If a customer has a custom segment under the *By campaign* facet dimension called *testSegment* with the guid *acdfb2fc-b2d2-452f-ac5a-6dcb5a3b9cae*, the customer must use this guid in the script that migrates data for Campaign facet metrics.

The following example illustrates this:

```
IF(OBJECT_ID('dbo.[Fact_CampaignMetrics]', 'U') IS NOT NULL)
BEGIN
INSERT INTO Fact_CampaignGroupMetrics
SELECT *
FROM Fact_CampaignMetrics
WHERE SegmentId = '2B6EF914-175F-4DC1-BE6F-05255B437A91';
INSERT INTO Fact_CampaignFacetMetrics
SELECT *
FROM Fact_CampaignMetrics
WHERE SegmentId IN ('F47B44D8-07E4-4CD5-A423-0EB7F8A2A3DE', 'acdfb2fc-b2d2-452f-ac5a-
6dcb5a3b9cae');
INSERT INTO Fact_CampaignFacetGroupMetrics
SELECT *
FROM Fact_CampaignMetrics
WHERE SegmentId = 'CA62842F-CBFE-4C7A-856E-4BB5E4FE30E1';
DELETE FROM Fact_CampaignMetrics
WHERE SegmentId IN('2B6EF914-175F-4DC1-BE6F-05255B437A91', 'F47B44D8-07E4-4CD5-A423-
0EB7F8A2A3DE', 'CA62842F-CBFE-4C7A-856E-4BB5E4FE30E1');
END;
```

Note
This is required for segments that are created under the Experience Analytics Dimensions.

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 *The xDB databases*.

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 10.0.0, 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 was 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 10.0.0 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 10.0.0. 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 `XConnect Upgrade Tool.zip` file. 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 recommended 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*, *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 *contactPercentile, contactPercentile,* and *contacts* 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
```