

# Upgrade Guide for Sitecore Experience Platform

Sitecore XP 10.1.0

*How to upgrade to Sitecore XP 10.1.0 from Sitecore 8.1.0 or later*

## Table of Contents

Chapter 1	Introduction .....	4
Chapter 2	Prepare for the upgrade.....	5
2.1	Get acquainted with the changes in Sitecore XP .....	6
2.1.1	Requirements .....	6
2.1.2	Release notes.....	6
2.1.3	Important changes.....	6
2.2	Prerequisites for upgrading .....	10
2.2.1	Backup the databases .....	10
2.2.2	Download files and the tools.....	10
2.2.3	Prepare your solution.....	10
2.2.4	Remove obsolete module content.....	10
2.3	Prepare xDB .....	11
2.3.1	Restore the analytics items for Sitecore XP 8.x .....	11
2.3.2	Stop the functionality that uses analytics .....	12
Chapter 3	Upgrade Sitecore XP .....	14
3.1	Safeguard the databases.....	15
3.2	Upgrade the databases.....	16
3.2.1	The XM databases .....	16
3.2.2	The xDB databases .....	17
3.2.3	Clean up the content databases .....	22
3.2.4	Configure any customized databases .....	23
3.3	Install and configure Sitecore XP 10.1.0 .....	26
3.3.1	Install a new instance of Sitecore XP .....	26
3.3.2	Connect to SQL databases .....	26
3.3.3	Sitecore Identity.....	26
3.3.4	Deploy your Sitecore solution .....	26
Chapter 4	Post-upgrade steps .....	27
4.1	General maintenance.....	28
4.1.1	Update personalization rules .....	28
4.1.2	xDB maintenance .....	29
4.2	Update xDB .....	30
4.2.1	Upgrading from Sitecore XP 8.x .....	30
4.2.2	Upgrading from Sitecore XP 9.0.1 .....	33
4.2.3	Upgrading from Sitecore XP 9.0.2 or earlier.....	34
4.2.4	Upgrading from Sitecore XP 9.1.1 or earlier.....	35
4.2.5	Upgrading from Sitecore XP 9.2.0 or earlier.....	35
4.2.6	Upgrading from Sitecore XP 10.0.1 or earlier.....	36
4.3	Rebuild the Reporting database .....	38
4.4	Migrate the xDB data .....	39
4.4.1	Use the xDB Data Migration tool to migrate xDB data .....	39
4.4.2	Migrate the xDB data.....	39
Chapter 5	Appendices.....	41
5.1	Sitecore recommends .....	42
5.1.1	Enable telemetry reporting in production environments .....	42
5.1.2	Azure Key Vault.....	43
5.1.3	Cache configuration.....	43
5.2	Appendix A – Scripts.....	44
5.3	Appendix B – Generating shard keys and missed percentile values.....	45

5.3.1	Run the Sitecore Xdb Collection Update tool .....	45
5.4	Appendix C - Update Experience Analytics custom segments .....	48

*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 © 2001-2022 Sitecore. All rights reserved.*

# Chapter 1

## Introduction

This guide describes how to upgrade to Sitecore Experience Platform 10.1.0 from Sitecore Experience Platform 8.1.0 or later.

The instructions in this guide are not valid for containerized solutions.

For more information about upgrading in containers, see the *Upgrade Container Deployment Guide*.

Starting with Sitecore XP 10.1.0, Sitecore uses a new upgrade methodology where you:

- Upgrade the databases.
- Install the new version of Sitecore XP beside your existing Sitecore XP installation.
- Attach the databases to the new installation.
- Deploy your solution on top of the new installation.
- Perform some post-upgrade steps.

Before upgrading your Sitecore XP production system, we recommend that you test the upgrade process on a test or developer system. This can highlight issues that may be caused by any custom tools and applications that you are using.

If you are upgrading from Sitecore XP 8.1.0 – 10.0.1, you can upgrade directly to Sitecore XP 10.1.0. However, if your solution is based on an earlier version of Sitecore, you must first upgrade to Sitecore XP 8.1.0 before you can upgrade to Sitecore XP 10.1.0.

## Chapter 2

# Prepare for the upgrade

This chapter describes the prerequisites that are required to upgrade to Sitecore XP 10.1.0 and the steps that you must perform before you start the actual upgrade process.

This chapter contains the following sections:

- Get acquainted with the changes in Sitecore XP
- Prerequisites for upgrading
- Prepare xDB

## 2.1 Get acquainted with the changes in Sitecore XP

### 2.1.1 Requirements

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

For more information about the requirements, see the [Sitecore XP 10.1.0 Installation Guide](#).

#### SQL Server

Before you upgrade the databases, you must ensure that your SQL Server installation is compatible with the new version of Sitecore XP.

If your current version of SQL Server is not compatible, you must upgrade to a compatible version of SQL Server before you upgrade or deploy the Sitecore databases.

For more information, see the [Sitecore compatibility table](#).

#### Solr

Ensure you have a compatible version of Solr. For more information, see the [Sitecore compatibility table](#).

When deploy the Sitecore XP 10.1.0 – XP Single topology – in a local development environment, Solr server and all the required cores and configurations are installed.

If you use SolrCloud, add the authentication for SolrCloud to the appropriate connection strings.

### 2.1.2 Release notes

Review the release notes for Sitecore XP 10.1.0 and the other releases that came between the version your are upgrading from and Sitecore XP 10.1.0:

- [Sitecore Experience Platform 10.1.0](#)
- [Sitecore Experience Platform 10.0.0](#)
- [Sitecore Experience Platform 9.3](#)
- [Sitecore Experience Platform 9.2](#)
- [Sitecore Experience Platform 9.1](#)
- [Sitecore Experience Platform 9.0](#)
- [Sitecore Experience Platform 8.2](#)

### 2.1.3 Important changes

This section describes some of the major changes that have been made since Sitecore XP 9.0.0. You must follow the upgrade recommendations in this section.

#### Sitecore XP 10.1.0

- The MongoDB provider is no longer supported.

For the *Collection* database you must use SQL Server.

You can use [xDB Data Migration Tool](#) to migrate the *Collection* data from MongoDB to SQL Server.

## Sitecore XP 10.1.0

For the Session state you must use the Redis session state provider or the SQL session state provider.

- The dedicated *Reporting* server role has been combined with the *Content Management* (CM) server role to reduce hosting costs. If you have made any customizations to the *Reporting* server role in previous versions, you must apply these to the *CM* server role.

There is no need to update the *Reporting* server role, it can be removed from your architecture/hosting.

### Sitecore XP 10.0.0

- The `Xdb.MarketingAutomation` connection string was removed and is no longer required on CM instances.
- Support for Azure Search was deprecated.
- The Experience Analytics configuration files were changed.

If you have enabled reaggregation before the upgrade, delete the old reaggregation configuration files. The reducer configuration files are now disabled by default. Make sure that the configuration files are not present in the `App_Config\Sitecore\ExperienceAnalytics` folder.

New versions of the reaggregation and reducer configuration files are in the `App_Config\Include\Examples` folder. Keep them disabled until the entire upgrade process is finished.

### Sitecore XP 9.3.0

- 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 the new configuration files in the `App_Config\Sitecore\CMS.Core\` folder.
- The Lucene search provider is no longer supported.
- All communication with the *Content Management* server occurs over HTTPS. If you don't yet use HTTPS, 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](#).
- The concept of execution strategies on the SQL provider level was introduced in xConnect.

There are two types of execution strategies:

- TVP
- Staging Tables.

xConnect is configured to use the *TVP* strategy by default. However, it is not compatible with the *Always Encrypted* feature. If you use the *Always Encrypted* feature, you must change the execution strategy to *Staging Tables*. For more information about configuring the execution strategies, [see the Sitecore documentation](#).

### Sitecore XP 9.2.0

- A new *Indexing* sub-role was introduced to simplify the configuration of content indexing. For more information about role-based configuration, see the [Sitecore Documentation](#).

### Sitecore XP 9.1.0

- The Sitecore configuration now supports the [Helix](#) architecture principles. The `Website\App_Config\Include` folder contains folders and files that are loaded in the following order:
  - *Foundation* folder
  - *Feature* folder
  - *Project* folder
  - The files in the `Website\App_Config\Include` folder.
- The `security` connection string became a required on the *CD*, *CM* and *Processing* roles. You can [move security](#) from the *Core* database to another database. If you do not intend to move security to another database, you must still ensure that the `security` connection string points to the *Core* database.
- Sitecore Identity (SI) mechanism for logging in to Sitecore was introduced. 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.

### Sitecore XP 9.0.1

- Sitecore Email Experience Manager (EXM) became a component of Sitecore XP. For more information, see:
  - [Configuring EXM in a scaled environment](#)
  - [Configure a dedicated email dispatch server](#)If you are upgrading from Sitecore XP 8.1 - 9.0.0 with the *Email Experience Manager* (EXM) module, which is no longer supported, and want to keep your data in the EXM component, you must manually migrate your EXM data to Sitecore XP 10.1.0 – it is not migrated during the upgrade. For more information about migrating EXM data, see the [Sitecore EXM Data Migration Guide](#).
- The Microsoft Visual C++ 2015 Redistributable is required. It might already be installed with Microsoft Windows. Without it, Sitecore XP will fail to start up with the message: *Could not load file or assembly 'ChilkatDotNet46.dll' or one of its dependencies. The specified module could not be found.* For more information, see <https://www.microsoft.com/en-us/download/details.aspx?id=53587>

### Sitecore XP 9.0.0

- [Sitecore Forms](#) became a component of Sitecore XP.

## Sitecore XP 10.1.0

- The format of all the databases changed and they are now distributed as *Data-tier Application Component Packages (DACPAC)*.
- Sitecore xConnect was introduced.

For more information, see [Getting Started with xConnect](#).

## 2.2 Prerequisites for upgrading

This section describes the steps you must follow to prepare for the upgrade.

### 2.2.1 Backup the databases

Before you start the upgrade, make a backup of your databases.

### 2.2.2 Download files and the tools

You must download the following files and tools:

- [Sitecore 10.1.0 rev. 005207 \(upgrade files\).zip](#).

On the downloads page, in the *Upgrade options* section, this file is called *Configuration files for upgrade*.

Unpack the `Sitecore 10.1.0 rev. 005207 (upgrade files).zip` file and then unpack:

- `Database Upgrade Script.zip` - contains SQL scripts to upgrade databases schema and data.
- `Databases.zip` - contains the DACPAC files that you need when deploy new database as part of this upgrade.
- `MarketingDefinitionsUpgrade.zip` - contains the `.aspx` pages that help you prepare your marketing definitions for the upgrade from Sitecore XP 8.x.
- [Sitecore.UpdateApp 1.1.0 for Sitecore X.X.X rev XXXXXX.zip](#)

You must download the package for the version of Sitecore XP that you are upgrading from. This file contains the clean-up tool.

### 2.2.3 Prepare your solution

To prepare for the upgrade, you must:

- Update the Nuget package references.
- Recompile your solution with the updated Sitecore DLLs and fix any breaking changes.
- Adjust the configuration patches for the new Sitecore version.
- Prepare the artifacts for deployment - containers images, WDP packages - including the modules that you plan to use.

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

### 2.2.4 Remove obsolete module content

If you use any Sitecore modules, you must find the corresponding version of the module and then refer to its documentation. You must only execute the steps that relate to the databases, like removing previous versions of items, to ensure a smooth upgrade of Sitecore XP.

## 2.3 Prepare xDB

If you are using the Sitecore Experience Database (xDB) functionality, you must complete the steps described in this section.

### 2.3.1 Restore the analytics items for Sitecore XP 8.x

#### Important

If you are upgrading from Sitecore XP 8.1.0 – Sitecore XP 8.2 Update 7, review this section.

If you are upgrading from Sitecore XP 9.0.0 or later, skip this section.

Before you upgrade, you must restore any of the standard marketing taxonomies and marketing definitions that you have deleted from your Sitecore installation.

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

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

Before you restore anything:

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

#### Important

The `RestoreDeletedMarketingTaxonomies.aspx` and `RestoreDeletedMarketingDefinitions.aspx` pages require direct access to the *Reporting* database. You must ensure that the `\App_Config\ConnectionStrings.config` file contains the appropriate connection string and that the database is accessible.

You can access these pages from a CM instance.

To restore deleted marketing taxonomy items:

1. To open the *Restore Deleted Marketing Taxonomies* page, enter the following URL in a browser:  
<http://<hostname>/sitecore/admin/RestoreDeletedMarketingTaxonomies.aspx>
2. Click **Restore deleted taxonomy items**.

To restore deleted marketing definition items:

1. To open the *Restore Deleted Marketing Definitions* page, enter the following URL in a browser:  
<http://<hostname>/sitecore/admin/RestoreDeletedMarketingDefinitions.aspx>
2. Click **Restore deleted definition items**.

All the default marketing taxonomy items and marketing definition items are restored from the *Reporting* database to the *Master* database. The following files are created in the `\App_Data` folder:

- `RestoredTaxonomies_{timestamp}.dat`
- `RestoredDefinitions_{timestamp}.dat`

These files list all the restored items. When the upgrade is completed, if you want to remove these items again, you can use these files to identify them.

## Validate the marketing definition names

The Marketing Operations API requires that the alias for each marketing definition is unique and by default consists only of letters, numbers, hyphens, and spaces or underscore characters.

The name of the marketing definition item is used as the value of the *Alias* property. The *Definition Items Name Validator* page validates the names of the marketing definition items so that the names can be edited before the upgrade.

In addition, you can make changes to the configuration and define your own rules for the permissible set of characters that can be used for aliases.

To validate the names of the marketing definition items:

1. To open the *Definition Items Name Validator* page, enter the following URL in a browser:

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

2. Click **Validate**.

The names of the definition items are validated, and a report called `DefinitionItemsNameValidationErrors_{timestamp}.dat` is saved in the `\App_Data` folder.

This report lists all the validation errors that occurred.

3. Ensure you manually fix all the issues that were found, and then validate the names again.

Repeat this procedure until all the names have been validated.

### 2.3.2 Stop the functionality that uses analytics

If you have any functionality that accesses the contact/interaction data extraction process from outside of xConnect, you must stop it. You can restart it when the upgrade is completed. An example of this kind of functionality is rebuilding the *Reporting* database.

#### Important

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

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

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.

### **Stop all content tests**

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

## Chapter 3

# Upgrade Sitecore XP

To upgrade to Sitecore XP 10.1.0, you must

- Be a local administrator on SQL Server.
- Upgrade your databases to ensure that they are compatible with the new version of Sitecore XP.
- Install the new version of Sitecore XP beside your existing Sitecore XP installation.
- Attach the databases to the new installation.
- Deploy your solution on top of the new installation.

This chapter contains the following sections:

- Safeguard the databases
- Upgrade the databases
- Install and configure Sitecore XP 10.1.0

### 3.1 Safeguard the databases

The upgrade process involves executing some SQL scripts to upgrade the databases. However, if you execute the scripts against databases that are in use, it can make the system unstable. To avoid this situation, you must safeguard the databases by ensuring that they are not being used when you run the SQL scripts.

To safeguard the databases, you can:

- Use the backup tools to make a copy of the databases and follow the instructions in this guide to upgrade these copies.

For more information about how to configure the *Collection* database after you copy it, see our [documentation](#).

Or

- Disable the environment and follow the instructions in this guide to upgrade the databases.

Stop the Sitecore web applications and services and turn off the application roles like CM or CD by stopping the sites in the IIS Management Console.

If you are upgrading from Sitecore XP 9.0.0 or later and using the xDB functionality, you must also stop the xConnect web application roles and the following windows services:

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

## 3.2 Upgrade the databases

Depending on which version of Sitecore XP you are upgrading from, you must upgrade some existing databases and deploy some new ones.

### Database upgrade

The database upgrade scripts contain the changes that must be applied to the Sitecore databases to align them with the new version. 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:

- 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.
- If you need to upgrade a database, extract the database upgrade scripts from the `Database Upgrade Script.zip` file that you downloaded earlier.

### Database deployment

In the new upgrade process, all the databases are installed when you install the new instance of Sitecore XP.

If you want to deploy the new databases manually, extract the appropriate DACPAC file from the `Databases.zip` file that you downloaded earlier. When you deploy a new database, ensure to update the connection string for the appropriate Sitecore role.

To restrict access to the database, you must create a user with limited access rights (a least privileged user) and specify this user in the connection string. If you do not 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*.

There are sample scripts for creating users for each database in the WDP packages.

### 3.2.1 The XM databases

The following table lists the databases that you must upgrade or deploy for the XM topology or for the XM part of the XP topology. The table also lists the scripts that you must run against the databases:

Database	Sitecore XP version			
	8.1.0 – 8.2.7	9.0.0 – 9.2.0	9.3.0	10.0.0 – 10.0.1
core	Scripts A, C	Scripts B, C	Script C	No action
master	Scripts A, D	Scripts B, D	Script D	No action
web	Scripts A, E	Scripts B, E	Script E	
experience forms	* Deploy	Script F, G	Script F	No action

\*The `ExperienceForms` connection string is required on *CD* and *CM* servers.

**Important**

The scripts listed in the table must be run against the databases in the order that they are listed in the table.

The scripts are:

- Script A - `CMS_core_master_web8x.sql`
- Script B - `CMS_core_master_web9x.sql`
- Script C - `CMS_core.sql`
- Script D - `CMS_master.sql`
- Script E - `CMS_web.sql`
- Script F - `SXP_experienceforms_storage.sql`
- Script G - `SXP_experienceforms_filestorage.sql`

**The Sessions database**

If you are using an out-of-process session state provider, you must perform the steps in this section.

- *SQL Server*  
Replace the *Session* database with a clean one from the Sitecore XP installation package.
- *Redis cache*  
Use the Redis Cache service *Console* to clean the cache.
- *Mongo*  
Switch to the SQL Server or Redis provider.

**3.2.2 The xDB databases**

To continue using the xDB functionality, you must follow the steps described in this section.

**Sitecore XP 8.x**

If you are upgrading from Sitecore XP 8.x, a number of new databases are installed with the new version of Sitecore XP.

**New xDB databases**

- processing tasks  
The `xdb.processing.tasks` connection string is required on the *Processing* server.
- exm\_master  
The `Exm.master` connection string is required on the *CM* servers.

**New xConnect databases**

- processing pools
- marketing automation
- collection shards

## Sitecore XP 10.1.0

- collection shard map manager
- reference data
- processing engine storage
- processing engine tasks
- messaging

To see which connection strings xConnect requires from the Sitecore server roles, review the connection strings on the new Sitecore installation.

When you have upgraded to Sitecore XP 10.1.0, you can use the xDB Data Migration Tool to migrate your data.

### Reporting database

You can use the xDB Data Migration Tool to migrate your data to the new version of the *Reporting* database.

Alternatively, you can execute the upgrade scripts to upgrade the *Reporting* database.

For more information about running the upgrade scripts, see the next section *Sitecore XP 9.0.0 and later*.

### Sitecore XP 9.0.0 and later

The following table lists the databases that you must upgrade or deploy for the Sitecore XP topology. For some databases the table lists the scripts that you must run against the xDB database. For other databases, you can find more details after the table.

Database	Sitecore version					
	9.0.0	9.0.1 – 9.0.2	9.1.0	9.1.1 – 9.2.0	9.3.0	10.0.0 – 10.0.1
processing engine storage	Deploy		Upgrade		No action	
processing engine tasks	Deploy		Upgrade			
messaging	Deploy	SXP_Messaging.sql*				
exm master	Deploy	Upgrade		No action		
processing tasks	SXP_processing_tasks.sql					No action
processing pools	SXP_processing_pools.sql					No action
reporting	Upgrade					No action
reference data	Upgrade				No action	
collection smm	Upgrade				No action	
collection shards	Upgrade					
marketing automation	SXP_marketingautomation.sql					

\*If you are using the Azure Service Bus, you do not need to upgrade the *messaging* database.

For newly deployed databases the following connection strings are added:

- `processing.engine.tasks` – the connection string is required on the *Processing engine* service.
- `processing.engine.storage` – the connection string is required on the *Cortex Processing*, and *Processing engine* services.
- `messaging` – the connection string is required on the *CM* and *CD* server roles and on the following xConnect roles:
  - `collection`
  - `collection search`
  - `marketing automation`
  - `automation engine service`
  - `processing engine service`
- `exm.master` – the connection string is required on the *CM* servers.

### Exm.master

To upgrade the *exm.master* database :

1. If you are upgrading from Sitecore XP 9.0.1 – 9.0.2, run the `SXP_exm_master_dispatchqueue.sql` script.
2. For every version that requires you to upgrade the *exm.master* database, run these scripts in the following order:
  - `SXP_exm_master_suppressions.sql`
  - `SXP_exm_master_save_suppressions.sql`
  - `SXP_exm_master_update_suppressions.sql`

### Collection – on SQL Server and SQL Azure

To upgrade the *Shard Map Manager* database:

- Run the `SXP_collection_smm.sql` script.  
This script registers the schemas for the shard maps.

To upgrade the *collection shards* database:

#### Note

The upgrade path for the *collection shards* database depends on the *Always Encrypted* (AE) feature and whether it is configured (On) on your *collection* database or not (Off).

Step 1:

You must perform this step.

- If AE is On:
  - Run the `SXP_collection_AE_Part1.sql` script for every shard.
  - 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 AE is Off:
  - Run the `SXP_collection_Part1.sql` script for every shard.
- If AE is On or Off:

In Sitecore XP 9.1.0, the `[UnlockContactIdentifiersIndex_Staging]` table was introduced. The collection user in every shard must be granted SELECT and INSERT permissions for this table.

To grant the collection user these permissions:

- 1) Open the `SXP_collection_GrantPermissions.sql` script and edit it to specify the user name in the following lines:

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

- 2) Run the script.

Step 2:

### Important

You *must* perform this step if you are upgrading from 9.0.0 - 9.2.0, or if you previously upgraded to Sitecore XP 9.3.0 or later and skipped this step

However, we recommended that you perform this step to align the database schema with the changes that were introduced in Sitecore XP 9.3.0. However, this step is optional

Additionally, in SQL Azure, the aligned schema will enable you to use the scale-out feature with the help of Microsoft Split/Merge service. For more information, see the article [Split or merge xDB Collection database shards](#).

1. To generate the shard keys for data rebalancing and/or missed percentile values, run the *Sitecore Xdb Collection Update* tool.

For more information, see *Appendix B – Generating shard keys and missed percentile values*.

2. After you have run the *Sitecore Xdb Collection Update* tool:

If AE is On:

- Run the `SXP_collection_AE_Part2.sql` script for every shard.

If AE is Off:

- Run the `SXP_collection_Part2.sql` script for every shard.

## Collection - on MongoDB

### Important

In Sitecore 10.1.0, the MongoDB provider is no longer supported. You must use the SQL server provider.

The *Collection* databases is deployed with new Sitecore XP installation. If you want to create a custom shard cluster, use the [SQL Sharding Deployment Tool](#).

## 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.

If you wish to retrieve the historical values with the accurate calculation, you can rebuild the reporting database after the upgrade process is complete.

### 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.

## xDB Reference Data

You must upgrade the *xDB Reference Data* database. We recommend that you make a copy of the database before you upgrade the database.

### Important

If you are upgrading from Sitecore XP 9.0.0 or later and it is deployed in the Azure App Service, you must [change the collation](#) of the *xDB Reference Data* database from `SQL_Latin1_General_CP1_CI_AS` to `Latin1_General_CS_AS` before you upgrade the database.

To upgrade the *xDB Reference Data* database:

1. Run the `SXP_referencedata_Part1.sql` script.

2. Review the results of running the 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.

3. Run the `SXP_referencedata_Part2.sql` script.

## 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.

### 3.2.3 Clean up the content databases

Database clean-up is a one-time operation that you must perform when you upgrade from any version of Sitecore XP to Sitecore XP 10.1.0.

In Sitecore XP 10.1.0, all the default items have been moved into resource files and you must remove these items from your databases.

The *Sitecore.UpdateApp* tool removes these items, if they have not been modified, from the *Core*, *Master*, and *Web* databases. If you have modified these items, they are not removed and the tool adds a list of them to its log file.

We recommend that you delete these items from the SQL database. When you have deleted the items, your Sitecore solution uses the new versions of these items that are in the resource files.

If you have modified any of these items, we recommend that you review them before you delete them to ensure that you still need these modifications. If you decide that you still need the modifications, re-apply your customizations to the new items after you have deleted the old ones.

The *Sitecore.UpdateApp* is a standalone console tool.

To clean up the content databases:

1. Locate the *Sitecore.UpdateApp 1.1.0 for Sitecore X.X.X rev XXXXXX.zip* file that you downloaded earlier and extract its contents to a folder, for example, `C:\Sitecore.UpdateApp`.
2. Copy the license file to the *Data* folder of the tool, for example, `C:\Sitecore.UpdateApp\Data\license.xml`.

- In the `App_config` folder, update the `ConnectionStrings.config` file with the connections to your databases, for example,  
`C:\Sitecore.UpdateApp\App_Config\ConnectionStrings.config`.  
 If you do not have a *security* database, use the connection to the *core* database.
- Open a Command Prompt in the tool folder and run:

```
Sitecore.UpdateApp.exe clean
```

The number of items that are removed from each database is listed in the Command Prompt window.

```
C:\Sitecore.UpdateApp.Content\content>Sitecore.UpdateApp.exe clean
Clean up resource items:
master: 5592 item(s)
web: 5591 item(s)
core: 19448 item(s)
Resource items are cleaned up.
C:\Sitecore.UpdateApp.Content\content>
```

Detailed information about any modified items that are skipped is written to the log files in the tool folder - `\Data\logs`.

#### Note

If you have more than one *Web* database, you must run steps 3 and 4 for every *Web* database.

You must update the connection string to each *Web* database as you run the tool, and not change the connection strings to the other databases.

## Troubleshooting

If the Command Prompt displays the following error:

*The type initializer for 'Sitecore.SecurityModel.License.LicenseManager' threw an exception.*

Verify that the correct license file is copied to the `Data` folder of the tool.

If the Command Prompt displays the following error:

```
System.InvalidOperationException: Connection string is empty. Name: '<database name>'
```

Verify that the `ConnectionStrings.config` file in the `App_config` folder of the `UpdateApp.Content` contains the correct connection strings.

If you do not use a separate *security* database, the security connection string must have the same value as the *core* connection string.

### 3.2.4 Configure a customized database

In Sitecore XP 10.1.0, if you have customized a database, you must use the Sitecore composite data provider to access the database. You must update your patch configuration file, and specify that the customized database must use the composite data provider.

For example, if you have a *web\_preview* database, you could use the following configuration patch file to update your configuration to use the composite data provider:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration xmlns:patch="http://www.sitecore.net/xmlconfig/">
  <sitecore>
```

```

<eventing defaultProvider="sitecore">
  <eventQueueProvider>
    <eventQueue name="web_preview" patch:after="eventQueue[@name='web']"
type="Sitecore.Data.Eventing.&(database)EventQueue, Sitecore.Kernel">
      <param ref="dataApis/dataApi[@name='&(database)']" param1="&(name)" />
      <param hint="" ref="PropertyStoreProvider/store[@name='&(name)']" />
    </eventQueue>
  </eventQueueProvider>
</eventing>
<PropertyStoreProvider>
  <store name="web_preview" patch:after="store[@name='web']" prefix="web_secondary"
getValueWithoutPrefix="true" singleInstance="true"
type="Sitecore.Data.Properties.&(database)PropertyStore, Sitecore.Kernel">
    <param ref="dataApis/dataApi[@name='&(database)']" param1="&(name)" />
    <param resolve="true" type="Sitecore.Abstractions.BaseEventManager, Sitecore.Kernel"
/>
    <param resolve="true" type="Sitecore.Abstractions.BaseCacheManager, Sitecore.Kernel"
/>
  </store>
</PropertyStoreProvider>
<databases>
  <!-- web_secondary -->
  <database id="web_preview" patch:after="database[@id='web']" singleInstance="true"
type="Sitecore.Data.DefaultDatabase, Sitecore.Kernel">
    <param desc="name">$(id)</param>
    <icon>Images/database web.png</icon>
    <securityEnabled>true</securityEnabled>
    <dataProviders hint="list:AddDataProvider">
      <dataProvider type="Sitecore.Data.DataProviders.CompositeDataProvider,
Sitecore.Kernel">
        <param desc="readOnlyDataProviders" hint="list">
          <protobufItems
type="Sitecore.Data.DataProviders.ReadOnly.Protobuf.ProtobufDataProvider, Sitecore.Kernel">
            <filePaths hint="list">
              <filePath>$(dataFolder)/items/$(id)</filePath>
              <modulesFilePath>sitecore modules/items/$(id)</modulesFilePath>
            </filePaths>
          </protobufItems>
        </param>
      <param desc="headProvider">
        <dataProvider ref="dataProviders/main" param1="$(id)">
          <disableGroup>publishing</disableGroup>
          <prefetch hint="raw:AddPrefetch">
            <sc.include file="/App_Config/Prefetch/Common.config" />
            <sc.include file="/App_Config/Prefetch/Webdb.config" />
          </prefetch>
        </dataProvider>
      </param>
    </dataProvider>
  </dataProviders>
  <PropertyStore ref="PropertyStoreProvider/store[@name='$(id)']" />
  <remoteEvents.EventQueue>
    <obj ref="eventing/eventQueueProvider/eventQueue[@name='$(id)']" />
  </remoteEvents.EventQueue>
  <archives hint="raw:AddArchive">
    <archive name="archive" />
    <archive name="recyclebin" />
  </archives>
  <cacheSizes hint="setting">
    <data>100MB</data>
    <items>50MB</items>
    <paths>2500KB</paths>
    <itempaths>50MB</itempaths>
    <standardValues>2500KB</standardValues>
  </cacheSizes>
</database>
</databases>
</sitecore>
</configuration>

```

You must also place your resource files in the appropriate items as resources folders.

1. In the `App_Data\items\` folder, make a copy of the `Web` folder.
2. Rename the copy of the folder and the `.dat` file that it contains.

Use the database name for the folder, for example, `web_preview` and use the database name in the file name, for example, `items.web_preview.dat`.

3. If you have SXA installed and if you have a `sitecore modules\items\` folder, in the `sitecore modules\items\` folder, make a copy of the `web` folder.
4. Rename the copy of the folder and the `.dat` file that it contains.

Use the database name for the folder, for example, `web_preview` and use the database name in the file name, for example, `items.web_preview.sxa.dat`.

## 3.3 Install and configure Sitecore XP 10.1.0

### 3.3.1 Install a new instance of Sitecore XP

To install Sitecore XP 10.1.0, follow the instructions in the [Installation Guide](#).

### 3.3.2 Connect to SQL databases

When the installation is finished, modify the `ConnectionStrings.config` file to connect the new Sitecore XP 10.1.0 instance to the databases from previous Sitecore XP version that you upgraded in the previous chapter.

Each server role must be configured appropriately.

### 3.3.3 Sitecore Identity

The latest version of Sitecore Identity server is installed with Sitecore XP 10.1.0.

Update the connection string to point it to the database you work with.

1. On the Sitecore 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 the users separately, you must edit the connection string for the *security* database.

You must recompile and redeploy any custom Sitecore identity plugins that you use on the new version.

If you are not using Sitecore Identity server, disable it in Sitecore XP 10.1.0.

### 3.3.4 Deploy your Sitecore solution

You should now have the new Sitecore XP 10.1.0 instance installed and connected to the upgraded databases.

Depending on your development process, you should:

- In your solution, reference the new Sitecore XP 10.1.0 components and assemblies.
- Fix the breaking changes and recompile your solution.
- Ensure that any configuration changes you have implemented are reconfigured in separate configuration patch files.
- Edit the `Web.config` file and any other non-patchable configuration files to meet your needs.
- Deploy your Sitecore solution on the new Sitecore XP 10.1.0 installation.

## Chapter 4

# Post-upgrade steps

After you have installed the new version of Sitecore XP and deployed your solution, you must complete some post-upgrade steps.

This chapter contains the following sections:

- General maintenance
- Update xDB
- Rebuild the Reporting database
- Migrate the xDB data

## 4.1 General maintenance

When you have finished the upgrade and upgraded any Sitecore modules that you use, 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.

- On the Sitecore Launchpad, open the **Control Panel** and:
  - In the **Indexing manager** rebuild the [content search indexes](#).
  - In the *Database* section, click **Rebuild link databases**, select the *Master* and *Core* databases, and then click **Rebuild**.
  - In the Database section, click **Clean up databases**, 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 client translations](#).

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

### 4.1.1 Update personalization rules

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>

## 4.1.2 xDB maintenance

If you are using xDB, you must:

- [Rebuild the xDB index.](#)
- 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**.

## EXM configuration

EXM is enabled by default, to use it you must have an EXM license. For more information about configuring email delivery, see [Sitecore Email Cloud compared to the custom SMTP](#).

You must use 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](#).

If you don't plan to use EXM you can disable it in the `\Website\Web.config` file, in the `<AppSettings>` section, setting the value to `"no"`:

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

## 4.2 Update xDB

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

The steps are listed according to the Sitecore version where the changes were introduced.

You must perform all the steps that are applicable for the version of Sitecore XP that you are upgrading from.

For example:

- If you have upgraded from Sitecore XP 9.0.2, you must perform all the steps that are applicable from Sitecore XP 9.0.2 through to Sitecore XP 10.0.1.

### 4.2.1 Upgrading from Sitecore XP 8.x

#### **Important**

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

#### **Delete unwanted marketing definitions and marketing taxonomies**

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

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

#### **Note**

If you try to remove any Sitecore items that are provided out of the box, they will not be removed. In Sitecore XP 10.1.0, these default items are supplied as resources and are immutable.

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

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

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

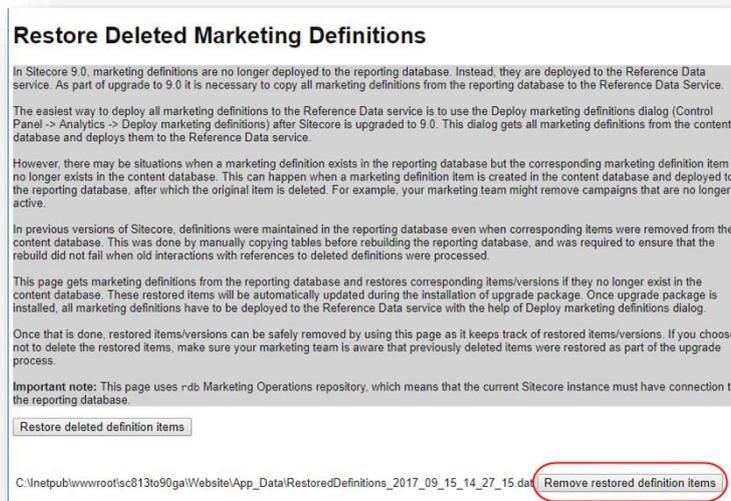
- In a text editor, open the `\App_Data\RestoredTaxonomies_{timestamp}.dat` file.

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

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

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

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

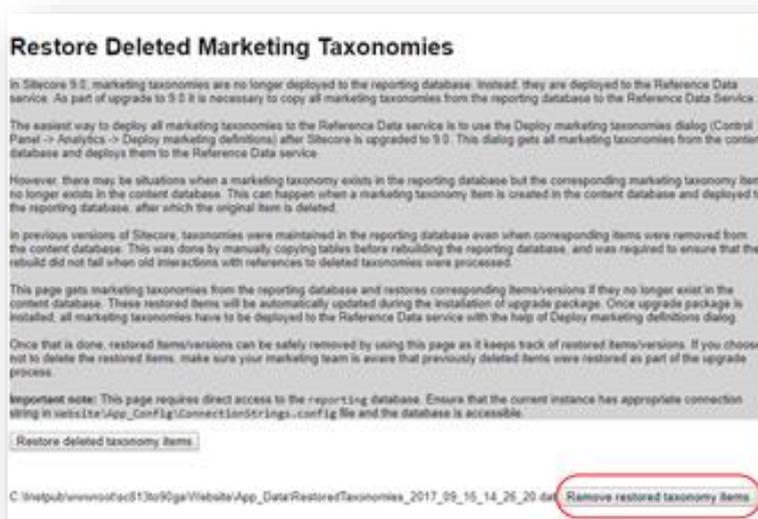


To delete unwanted marketing taxonomy items from the *Master* database:

- To open the *Restore Deleted Marketing Taxonomies* page, enter the following URL in a browser:

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

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



## Update customizations for reporting

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

All the filters and maps that come with the new version of Sitecore XP have been updated already.

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

## Upgrade Experience Analytics

In Sitecore XP 9.0.0 or later, some subsystems that affect Experience Analytics have been changed. If you are upgrading from Sitecore XP 8.x and have customized Experience Analytics, you must update these customizations to support the updated subsystems.

### Custom segments

If you have any custom filters or any segments that use these custom filters, you must update these filters manually and redeploy the segments.

The Rule Set Editor has been updated and the rules that it contains have also been changed.

You must therefore:

- Reconfigure any filters that use these custom segments and then redeploy the custom segments.
- Rebuild any custom filters that you have created so that they are compatible with the new subsystems.

For more information about editing filters, see the topic [Update a custom filter to use new rules](#).

### Custom reports

In Sitecore XP 9.0.0 or later, you can collect specific metrics for each dimension. If you created any custom reports or dashboards, you must update these manually, change the dimension/segment that it uses, and select the new version of the metrics that you want to use in the reports or dashboards.

For more information about adding a parameter to a report, see the topic [Experience Analytics](#).

## Upgrade the Path Analyzer

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

### Update custom maps

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

### Redeploy the Path Analyzer Maps

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

To redeploy the Path Analyzer maps:

1. In SQL Server Management Studio, in the *Reporting* database, to remove all the data from the *TreeDefinitions* table, run the following command:

```
Delete from TreeDefinitions
```

- In Sitecore, open the Path Analyzer admin page – `/sitecore/admin/pathanalyzer.aspx`.
- In the **Maps Manager** section, click **Deploy all maps that are not deployed**.

## 4.2.2 Upgrading from Sitecore XP 9.0.1

### Upgrade the contact merge data

#### Important

You must only follow this step if you are upgrading from Sitecore XP 9.0.1.

In Sitecore XP 9.0.2, the contact merge behavior was changed and when you merge contacts, interactions are copied from the source contact to the target contact and the source contact is marked as obsolete.

To apply the new contact merge logic to existing contacts that have previously been merged:

- On the xDB Processing server, open the `/sitecore/admin/xdb/ContactsMerge.aspx` page.



- Click **Start**.

This copies all the interactions from the source contact to the target contact and significantly improves the performance of data retrieval.

When the contact merge upgrade process is complete, the Process State changes to *Completed*.

#### Note

This process might take a long time and you can run it in a production environment while the system is live. You can also postpone this step, but it is an important part of the upgrade process and you *must* perform it.

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

After the merge process completes, summary information is written to the `contactMergeDataUpgrade.log.{date}.txt` file in the `logs` folder. The log file includes status and error information about the merge process. The log file is always created even when no information is written to it.

The administration page at `\Website\sitecore\admin\xdb\ContactsMerge.aspx` reports the errors noted in the log file.

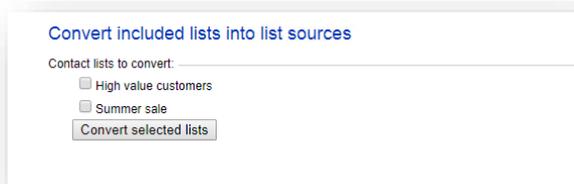
## 4.2.3 Upgrading from Sitecore XP 9.0.2 or earlier

### Upgrade the included lists in List Manager

In Sitecore XP 9.0.2 or earlier, you could add contact lists to other lists.

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

1. On the *CM* 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

#### Important

You must only follow this step if you are upgrading from Sitecore XP 9.0.0, 9.0.1, or 9.0.2.

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 *Appendix C - 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.

#### 4.2.4 Upgrading from Sitecore XP 9.1.1 or earlier

##### Important

If you are upgrading from Sitecore XP 9.1.1 or earlier, you must perform the steps in this section.

To update the `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 agents 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.

#### 4.2.5 Upgrading from Sitecore XP 9.2.0 or earlier

##### Important

If you are upgrading from Sitecore XP 9.2.0 or earlier, you must perform the steps in this section.

##### Remove deprecated Path Analyzer maps from the *Reporting* database

In Sitecore XP 9.3.0, the *First Visits* and *Returning Visits* maps were 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')
```

##### Customize the `MarketingDefinitionAliasValidationRegex` application setting

If you customized the alias validation expression, you must also apply the same customized expression to the `MarketingDefinitionAliasValidationRegex` application setting of both the CMS and xConnect websites after the upgrade:

```
<appSettings>
  <add key="MarketingDefinitionAliasValidationRegex" value="^[a-zA-Z0-9_-\x20]*$" />
</appSettings>
```

For more information about this step, see the section *Validate the names of the marketing definition items*.

## 4.2.6 Upgrading from Sitecore XP 10.0.1 or earlier

### Important

If you are upgrading from Sitecore XP 10.0.1 or earlier, you must perform the steps in this section.

### Customize pattern matching

In Sitecore XP 10.1.0, we introduced a new setting:

- `Marketing.Patterns.MinimalProfileScoreCount`

This setting replaces the `Analytics.Patterns.MinimalProfileScoreCount` setting in the `/Sitecore/Marketing/Tracking/Sitecore.Analytics.Tracking.config` file.

If you customized pattern matching by editing the value of the old

`Analytics.Patterns.MinimalProfileScoreCount` setting, you must also apply the same customizations to the new `Marketing.Patterns.MinimalProfileScoreCount` setting after the upgrade.

### Important

We recommend that you use a patch file to edit this setting.

Furthermore, on every xConnect instance, in the `sc.Marketing.XConnect.xml` configuration file, you must synchronize this new setting with the `ProfileDefinitionManagerSettings-IProfileDefinitionManagerSettings\Options\MinimalProfileScoreCount` setting.

### Deactivate unused segment definitions in the Reference Data database

In Sitecore XP 10.1.0, if you delete a segment definition, it is automatically deactivated in the *Reference Data* database.

If you are using the 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**.

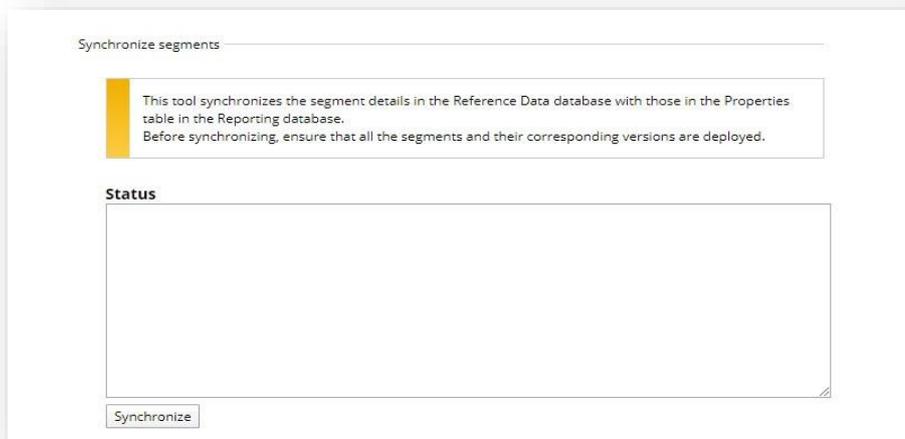
### Synchronize the Experience Analytics segments

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

In Sitecore XP 10.1.0, segment details are distributed between the *Reference Data* and *Reporting* databases. *Aggregation*, *Reaggregation*, and *Reducer* will not work on segments that are not synchronized across the *Reference Data* and *Reporting* database

To synchronize the segments across the *Reference Data* and *Reporting* databases:

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



If you are upgrading from Sitecore XP 9.2.0 or earlier, and you have created custom segments that uses filters, you must update the system dimension.

To update the system dimension:

1. On a CM instance, open the `/sitecore/admin/ExperienceAnalytics.aspx` page.
2. Click **Update System Dimension**



### **4.3 Rebuild the Reporting database**

After you have upgraded Sitecore XP, 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](#).

## 4.4 Migrate the xDB data

### Important

If you are upgrading from Sitecore XP 8.1.0 – 8.2.7, you must perform this step.

If you are upgrading from Sitecore XP 9.0.0 – 9.0.1, skip this step.

In Sitecore XP 9.0.0, we introduced a new data model for the Experience Database (xDB). This means that you must convert all the data in the MongoDB database to the new format. Use the xDB Data Migration Tool to perform this data conversion.

If you want to use the Sitecore Experience Database (xDB), and you do not want to keep your old data, you do not have to use the xDB Data Migration Tool to migrate your data.

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

### Note

The xDB Data Migration Tool only supports migrating data to Sitecore XP 10.0.1

If you are upgrading from Sitecore XP 8.1.0 – 8.2.7 and want to keep your xDB data, you must first upgrade to Sitecore XP 10.0.1. You can then use the [xDB Data Migration Tool](#) to migrate your data.

After you have migrated your xDB data to Sitecore XP 10.0.1, you can upgrade to Sitecore XP 10.1.0.

### 4.4.1 Use the xDB Data Migration tool to migrate xDB data

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

To deploy the new *Reporting* database:

1. Deploy the `Sitecore.Reporting.dacpac`.

This file is in the `Sitecore 10.0.1 rev. 004842 (upgrade files).zip` package, in the `\Databases` folder.

2. In the `ConnectionStrings.config` file, replace the name of the old reporting database with the name of the new one. For example:

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

In a non-scaled topology, update the `Reporting` connection string in the following files:

- o `sitecore\App_Config\ConnectionStrings.config`
- o `xconnect\App_Config\ConnectionStrings.config`
- o `xconnect\App_Data\jobs\continuous\ProcessingEngine\App_Config\ConnectionStrings.config`

In a scaled topology, update the connection strings for every *Content Management*, *Reporting* and *Cortex Reporting* instance.

### 4.4.2 Migrate the xDB data

After you deploy the new *Reporting* database to SQL Server, you can run the [xDB Data Migration tool](#).

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

## Chapter 5

### Appendices

This chapter contains the following sections:

- Sitecore recommends
- Appendix A – Scripts
- Appendix B – Generating shard keys and missed percentile values

## 5.1 Sitecore recommends

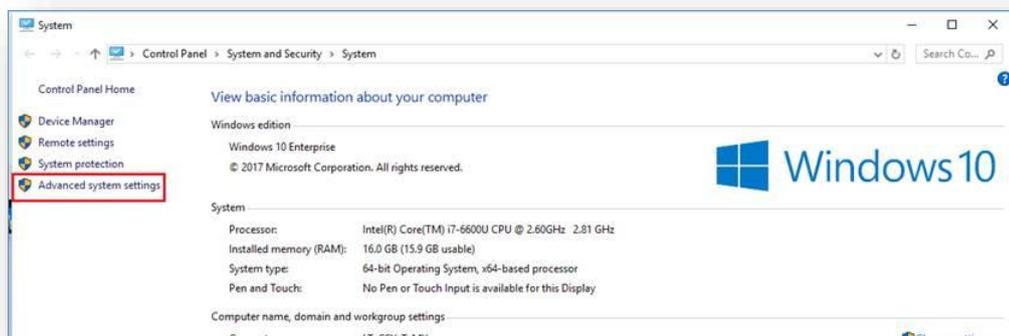
### 5.1.1 Enable telemetry reporting in production environments

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.

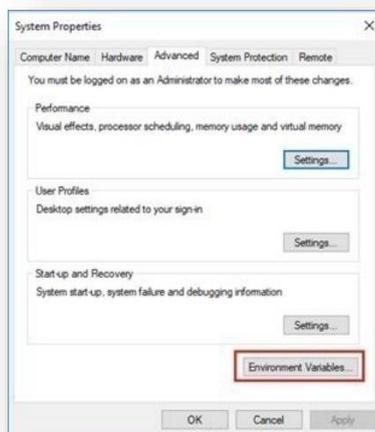
The telemetry system assumes Sitecore is running in a non-production environment by default. When you set up a production environment, you must therefore define a system variable to report the environment type.

To define a system variable:

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

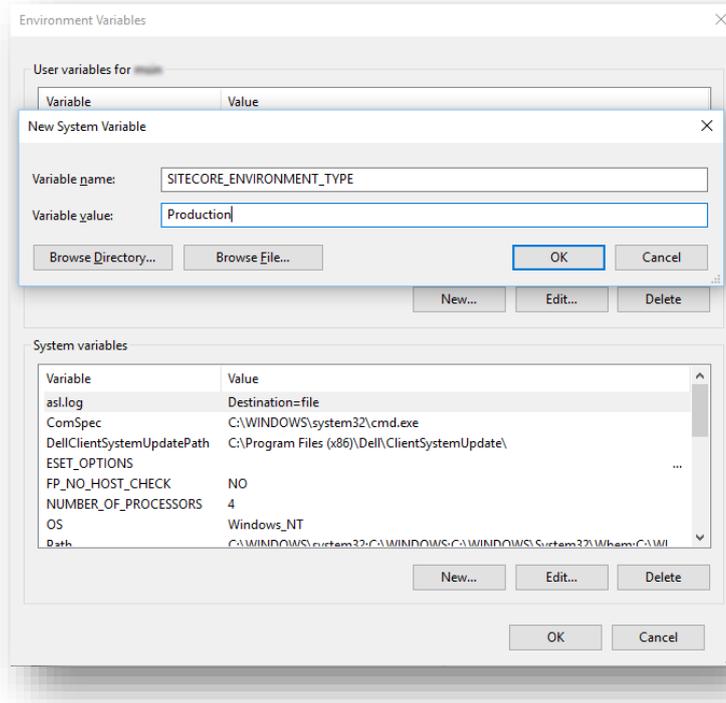


2. In the **System Properties** dialog, click **Environment Variables**.



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

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



- Restart the computer.

## 5.1.2 Azure Key Vault

In Sitecore XP, you can protect sensitive data with *Always Encrypted* on SQL Server and store the encryption keys in [Azure Key Vault](#).

For more information about how to configure Azure Key Vault, see [Sitecore's documentation](#).

For more information about Azure Key Vault, see [Microsoft's documentation](#).

## 5.1.3 Cache configuration

### Note

If you have upgraded from any version of Sitecore XP 8.1, read this section, otherwise skip it.

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

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

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

## 5.2 Appendix A – Scripts

### 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
```

## 5.3 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 about the xDB *Collection* database for the upgrade path you are following.

When you update the SQL *Collection* schema to version 10.1.0, the shard key values are null by default. Previous upgrades to version 9.2 might result in percentile value being less than 0.

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.

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).

### 5.3.1 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.

The Split-Merge functionality is only available if you are running Sitecore XP 10.1.0 on Azure (not on-Prem), and the *AlwaysEncrypted* feature is not configured. 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 10.1.0. You can use it to update production databases while only placing a minimal load on the database server.

The tool is stored in the `XConnect Upgrade Tool.zip` package that is part of the `Sitecore Sitecore 10.1.0 rev. 005207 (upgrade files).zip` package that you downloaded 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](#).

#### 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.

## 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.

## Tool arguments

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

Argument (short synonym)	Required	Default value	Description
<code>--cmd (-c)</code>	Yes		The update operation that you want to run.
<code>--connectionString (-s)</code>	Yes		The connection string to the collection shard, including the user ID and password.
<code>--consumersNumber (-cn)</code>	No	8	The number of threads that will send update queries.
<code>--batchSize (-bs)</code>	No	500	The number of entities that should be updated in a single query.
<code>--timeout (-t)</code>	No	30	The command query timeout.

<code>--startId (-si)</code>	No	00000000 -0000- 0000- 0000- 00000000 0000	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
```

## 5.4 Appendix C - 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.