SITECORE®

Upgrade Guide for Sitecore Experience Platform

Sitecore XP 10.3.0

How to upgrade to Sitecore XP 10.3.0 from Sitecore 8.1.0 or later

Table of Contents

Sitecore XP 10.3.0

Chapter 1

Introduction

This guide describes how to upgrade to Sitecore Experience Platform 10.3.0 from Sitecore Experience Platform 8.1.0 or later.

The instructions in this guide are only valid for on-premise solutions.

Since the release of Sitecore XP 10.1.0, Sitecore uses a new upgrade methodology where you:

o   Upgrade the databases.

o   Install the new version of Sitecore XP beside your existing Sitecore XP installation.

o   Attach the databases to the new installation.

o   Deploy your solution on top of the new installation.

o   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 or later you can upgrade directly to Sitecore XP 10.3.0. 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 10.3.0.

Chapter 2

Prepare for the upgrade

This chapter describes the prerequisites that are required to upgrade to Sitecore XP 10.3.0 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

- Safeguard the databases

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 10.3.0 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 Solr compatibility table.

When deploying Sitecore XP 10.3.0 – XP Single topology – in a local development environment, the 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.

xDB data migration from Sitecore XP 8.x

Important
If you are upgrading from Sitecore XP 8.1.0 – 8.2.7, you must read this section. If you are upgrading from Sitecore XP 9.X or later, skip this step.

In Sitecore XP 9.0.0, we introduced a new data model for the Experience Database (xDB). You must therefore use the xDB Data Migration Tool to convert all the data in the MongoDB database to the new format. The xDB Data Migration Tool only supports migrating data to Sitecore XP 10.0.1.

You can choose one of these options:

- If you do not want to keep your xDB data, you do not have to use the xDB Data Migration Tool. You can continue using Sitecore Experience Database (xDB) without your xDB data. Continue with the current guide.

- If you want to keep your xDB data, you must first upgrade to Sitecore XP 10.0.1 and 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 the latest Sitecore XP. If you select this option, you should now read the Sitecore XP 10.0.1 upgrade guide.

MongoDB

The MongoDB provider is no longer supported for the Collection database. If you use MongoDB now and want to upgrade to a later version of Sitecore, please familiarize yourself with the data migration options that are available.

2.1.2   Release notes

Review the release notes for Sitecore XP 10.3.0 and the other releases that came between the version your are upgrading from and Sitecore XP 10.3.0. Release notes are available for each release on the [Sitecore Experience Platform Download](#) site.

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 10.3.0

Sitecore Identity Server switched from IdentityServer4 to Duende IdentityServer under a commercial license.

Sitecore Identity Server requires [.NET Core 6.0 Windows Hosting Bundle](#).

Sitecore XP 10.2.0

- Sitecore XP no longer supports Azure Search.

  Solr is the only search provider that Sitecore XP supports.

Sitecore XP 10.1.0

- The MongoDB provider is no longer supported for the *Collection* database or as the 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.

- 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:

  o TVP

  o 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:

  o *Foundation* folder

  o *Feature* folder

  o *Project* folder

  o 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:

- o [Configuring EXM in a scaled environment](#)

- o [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 10.3.0 – 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.

- 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 10.3.X rev. 008463 (upgrade files).zip.

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

  Unpack the Sitecore 10.3.0 rev. 008463 (upgrade files).zip file and then unpack:

  o   Database Upgrade Script.zip – contains the SQL scripts that upgrade the database schemas and data.

  o   MarketingDefinitionsUpgrade.zip – contains the .aspx pages that help you prepare your marketing definitions for the upgrade from Sitecore XP 8.x.

- Sitecore.UpdateApp 1.3.X for Sitecore X.X.X rev XXXXXX.zip

  If you are upgrading from Sitecore XP 10.1.X or later and your solution does not contain any modules or connectors, you do not need this tool.

  If you are upgrading from Sitecore XP 10.1.X or later and your solution contains some modules or connectors, you need this tool for both the modules and the connectors.

  If you are upgrading from Sitecore XP 10.0.X and earlier, you need this tool for the Platform. If your solution contains any modules or connectors, you also need this tool for the modules and the connectors.

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

- You must upgrade to or install .NET Core 6.0 Windows Hosting Bundle or later.

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 pre-upgrade steps that are described in the upgrade guide for each module.

You can see a list of compatible modules on the downloads page for 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:

- o `RestoredTaxonomies_{timestamp}.dat`
- o `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 process was changed in Sitecore XP 9.1.0.

If you are upgrading from Sitecore XP 9.1.0 and later, you can safely stop and restart these tasks.

Processing Engine

If you are upgrading from Sitecore XP 9.1.0 or later, before you start the upgrade, we recommend that you review the list of tasks in the *ProcessingEngineTasks* database, in the *Tasks* table and:

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

- Stop scheduling new processing tasks.

- Ensure that there are no more tasks in the queue.

  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 Ensure that any deferred tasks that are in the *InProgress* state are re-run.

  o Ensure that 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.

2.4 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:

  o The Sitecore XConnect Search Indexer service

  o The Sitecore Marketing Automation Engine service

  o Sitecore Cortex Processing Engine service – if you are upgrading from Sitecore XP 9.1.0 or later

Chapter 3

Upgrade Sitecore XP

To upgrade to Sitecore XP 10.3.0, you must

o   Be a local administrator on SQL Server.

o   Upgrade your databases to ensure that they are compatible with the new version of Sitecore XP.

o   Install the new version of Sitecore XP beside your existing Sitecore XP installation.

o   Attach the databases to the new installation.

o   Deploy your solution on top of the new installation.

This chapter contains the following sections:

- Upgrade the databases

- Install and configure Sitecore XP 10.3.0

3.1 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.

Upgrade the existing databases

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.

Deploy the new databases

As part of the upgrade process, you install a new instance of Sitecore XP, where all the databases are deployed with the correct connection strings and with least privileged users.

If you want to deploy the new databases manually, extract the appropriate DACPAC file from the WDP files. When you deploy a new database, you must also update the connection string for the appropriate Sitecore role and restrict access to the database.

Restrict access to the database

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.

Since Sitecore XP 10.1.0 all the databases are deployed with least privileged users by default.

3.1.1   Clean up the content databases

Note
If you are upgrading from Sitecore XP 10.1.0 or later and your solution does not contain any modules or connectors, skip this section.

Because all the default items have been moved into resource files, you must remove them from your databases.

Database clean-up is a one-time operation.

The *Sitecore.UpdateApp* tool removes the default items, if they have not been modified, from the *Core*, *Master*, and *Web* databases.

If you have modified any of the default items, they are not removed and the tool adds a list of these items to its log file. We recommend that you review this list and decide if you still need the modifications. If you still need the modifications, re-apply your customizations to the new items after you have deleted the old ones. If you decide that you don't need the modifications, just delete these items from SQL databases.

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

To clean up the content databases:

1. On the Sitecore Launchpad, open the **Control Panel**, in the **Database** section, click **Clean up databases**, select all the databases, and then click **Clean**.

2. Locate the `Sitecore.UpdateApp 1.3.X 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`.

3. Copy the license file to the *Data* folder of the tool, for example, `C:\Sitecore.UpdateApp\Data\license.xml`.

4. In the `C:\Sitecore.UpdateApp\App_Config\ConnectionStrings.config` file, update the connections to your databases.

   If you do not have a *security* database, use the connection to the *core* database.

5. Add the upgrade resources from every module and connector to the UpdateApp Tool files.

   Note
   If your solution does not contain any modules or connectors, skip this step.

   You can download the upgrade packages for the different versions of the Sitecore modules from the Sitecore downloads site and you can find links to the compatible modules on the Sitecore XP downloads page.

   1) Download the items as resources zip file for the module version that is installed on your solution. If the module or the connector has several versions, there are separate folders for each version.

   2) Unpack the zip file into a local folder, for example, `c:\ModulesUpgradeResources\[Module Name]`.

   3) Copy all the subfolders and files from the *Data* folder, for example, from `c:\ModulesUpgradeResources\[Module Name]\X.X.X\Data`.

4) Paste all the subfolders and files into the *UpdateApp Data* folder, for example, `C:\Sitecore.UpdateApp\Data`.

5) Repeat this procedure for every module and connector.

6. 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`.

7. To clear the caches, open the `<instance_url>\sitecore\admin\cache.aspx` page, click **Refresh**, and then click **Clear all**.

8. If you have modified items, you should review and delete them after you install the new Sitecore instance and connect it to upgraded databases, as described in section 3.2.2 Connect to SQL databases under the "Complete databases cleanup" heading.

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.

When you clean up the databases on SQL Azure, you can get the following error:

A transport-level error has occurred when receiving results from the server. (provider: TCP Provider, error: 0 - The specified network name is no longer available.)

If this happens, repeat step 4 and run `Sitecore.UpdateApp.exe` again.

3.1.2 The XM databases

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

Database	Sitecore XP version		
	10.0.X	10.1.X	10.2.X
core	Script H*		Script H*
master	No action		No action
web	Script E	No action	No action
experience forms	No action		No action
security*	Script H*		Script H*

Database	Sitecore XP version		
	8.1.X – 8.2.X	9.0.X – 9.2.X	9.3.0
core	Scripts A, C, H*	Scripts B, C, H*	Scripts C, H*
master	Scripts A, D	Scripts B, D	Script D
web	Scripts A, E	Scripts B, E	Script E
experience forms	New**	Scripts F, G	Script F
security*	Not supported	Script H*	

* If you are not using the *security* database and the connection strings for the *core* and *security* databases are the same, you must apply script H to the *core* database.

If you have extracted the *security* database from the *core* database and the connection strings for *core* and *security* databases are different, you must apply script H only to the *security* database.

For more information about the *security* database, see [Walkthrough: Moving security data to a separate database](#).

** All the new databases are installed when you install the new instance of Sitecore XP.

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

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

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`
- Script H – `CMS_security.sql`

The Sessions database

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

- *SQL Server:* Use the new clean *Sessions* database from the Sitecore XP installation.
- *Redis cache:* Use the Redis Cache service *Console* to clean the cache.
- *Mongo:* Switch to the SQL Server or Redis provider.

3.1.3   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, Sitecore xConnect and a number of new databases are installed with the new version of Sitecore XP:

- New xDB databases
  - processing tasks
  - exm_master
- New xConnect databases
  - processing pools
  - marketing automation
  - collection shards
  - collection shard map manager
  - reference data
  - processing engine storage
  - processing engine tasks
  - messaging

To see which connection strings are required for the new xDB and xConnect databases, review the connection strings on the new Sitecore XP installation.

Reporting database for Sitecore 8.x

According to the 2.1.1 Requirements section, you should have chosen what to do with xDB data that you have in MongoDB, as MongoDB is no longer supported since Sitecore XP 9.0.0.

If you opted to not migrate your data, you should execute the upgrade scripts to upgrade the *Reporting* database as described in the next section. The new xConnect databases will not contain your data.

Sitecore XP 9.0.0 and later

The following tables list the databases that you must upgrade for the Sitecore XP topology. For some databases, the tables list the scripts that you must run against the databases. For other databases, you can find more details after the tables.

Database	Sitecore XP version		
	10.0.X	10.1.X	10.2.X
processing engine storage	No action		
processing engine tasks	`SXP_processing_engine_tasks.sql`	No action	
messaging	`SXP_Messaging.sql*`		No Action
exm master	No action		
processing tasks	No action		
processing pools	No action		
reporting	Upgrade		
reference data	No action		
collection smm	No action		
collection shards	Upgrade	No action	
marketing automation	`SXP_marketingautomation.sql`	No action	

Database	Sitecore XP version				
	9.0.0	9.0.1 – 9.0.2	9.1.0	9.1.1 – 9.2.0	9.3.0
processing engine storage	New**		`SXP_processing_engine_storage.sql`		No action
processing engine tasks	New**		`SXP_processing_engine_tasks.sql`		
messaging	New**	`SXP_Messaging.sql*`			
exm master	New**	Upgrade		No action	
processing tasks	`SXP_processing_tasks.sql`				No action
processing pools	`SXP_processing_pools.sql`				No action
reporting	Upgrade				
reference data	Upgrade				No action
collection smm	Upgrade				No action
collection shards	Upgrade				

Database	Sitecore XP version				
	9.0.0	9.0.1 – 9.0.2	9.1.0	9.1.1 – 9.2.0	9.3.0
marketing automation	SXP_marketingautomation.sql				

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

After you execute `SXP_Messaging.sql`, the *messaging* database is no longer aligned with a clean Sitecore XP 10.3.0 installation. The following tables are still present, but no longer in use:

- [dbo].[Sitecore_CT_ModelTrainingTaskStatusSubscriber]

- [dbo].[Sitecore_PE_TaskCancellationConsumer]

- [dbo].[Sitecore_PE_TaskCancellationProducer]

- [dbo].[Sitecore_PE_TaskProgressConsumer]

- [dbo].[Sitecore_PE_TaskProgressProducer]

- [dbo].[Sitecore_PE_TaskRegistrationConsumer]

- [dbo].[Sitecore_PE_TaskRegistrationProducer]

- [dbo].[Sitecore_PE_TaskStatusPublisher]

This ensures backward compatibility if you need to rollback to a previous version. Alternatively, you can remove them when you are confident that you do not need to roll back to a previous version.

** All the new databases are installed when you install the new instance of Sitecore XP.

The following connection strings are added for the newly deployed databases:

- *processing.engine.tasks* – the connection string is required on the *Processing engine* service and on the *Cortex Processing* 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
- *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.X, run the `SXP_exm_master_dispatchqueue.sql` script.

2. If you are upgrading from Sitecore XP 9.0.X - 9.1.0, run these scripts in the following order:

  o `SXP_exm_master_suppressions.sql`

  o `SXP_exm_master_save_suppressions.sql`

  o `SXP_exm_master_update_suppressions.sql`

Reporting

To upgrade the *Reporting* database, follow the steps below:

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

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.

3. Rebuild the *Reporting* database

   If you are upgrading from Sitecore XP 9.0.0 - 9.2.0, we recommended that you Rebuild the *Reporting* database when the upgrade process is complete. For more information see the section *Rebuild the Reporting database*.

   In Sitecore XP 9.3.0, two new columns – *Converted* and *FirstImpressionCount* - were added to the *Reporting* database artifacts to improve the accuracy of conversion and bounce rate calculations. If you wish to retrieve the historical values with accurate calculations for these columns – rebuild the *Reporting* database.

   If you decide not to rebuild the *Reporting* database, after the `SXP_reporting.sql` script execution it will set the default value of the conversion and the bounce rate of old data to zero and these values will only be calculated for new interactions after the upgrade.

xDB Reference Data

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.

Collection

> **Important**
>
> Since Sitecore XP 10.1.0, the MongoDB provider is no longer supported. You must use the SQL server provider. Familiarize yourself with the data migration options that are available.
>
> The SQL *Collection* databases are deployed with the new Sitecore XP installation. If you want to create a custom shard cluster, use the SQL Sharding Deployment Tool.

This section describes how to upgrade SQL *Collection* databases.

To upgrade the *Shard Map Manager* database:

- Run the `SXP_collection_smm.sql` script.

  This script registers the schemas for the shard maps.

You must also upgrade the *collection shards* databases.

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

To upgrade the *collection shards* database:

- Run the upgrade script for every shard:

  o If AE is On, run the `SXP_collection_AE_Part1.sql` script.

  o If AE is Off, run the `SXP_collection_Part1.sql` script.

- If you are upgrading from Sitecore XP 9.0.x and AE in On, grant permissions and configure encryption for the new *[UnlockContactIdentifiersIndex_Staging]* table.

  In Sitecore XP 9.1.0, the new *[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:

  o 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]
  ```

  o Run the script.

  If AE is On, in the new *[UnlockContactIdentifiersIndex_Staging]* table, configure deterministic encryption for the *[Identifier]* and *[Source]* columns, for every shard. To simplify the configuration process, SQL Server Management Studio provides an Encryption Wizard.

3.2 Install and configure Sitecore XP 10.3.0

3.2.1 Install a new instance of Sitecore XP

To install Sitecore XP 10.3.0, follow the instructions in the Installation Guide.

To install Sitecore modules follow the installation instructions for each module.

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

> **Note**
> If you are not using Sitecore Identity server, disable it in Sitecore XP.

3.2.2 Connect to SQL databases

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

You must configure each server role and Sitecore Identity Server appropriately.

Complete databases cleanup

If you have a list of modified items in the UpdateApp logs (as described in in section 3.1.1), you can now review and remove them from the databases, and re-apply the modifications that you still need.

After clenaup, Sitecore will function using default items from resource files, only further items modifications will be stored in databases.

3.2.3 Import client languages

If you had been using a client language other than English before the update, you must download and import the latest client translations following the Sitecore Experience Platform Client Translations Manual.

3.2.4 Deploy your Sitecore solution

You should now have the new Sitecore XP 10.3.0 instance installed and connected to the upgraded databases.

Depending on your development process, you should:

- In your solution, reference the new Sitecore XP 10.3.0 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 XP solution on the new Sitecore XP 10.3.0 installation.

3.2.5   Configure a customized database

If you are upgrading from Sitecore XP 10.0.X or earlier and 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.

See an example of a configuration patch file in [Appendix D – Example of configuration patch file](#).

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

4.1 General maintenance

When you have finished the upgrade and upgraded any Sitecore modules that you use, and before you start using Sitecore XP, you must:

- Clear the browser cache.

- Republish your website to every publishing target.

- [Rebuild all content search indexes](#).

- [Rebuild the link databases](#).

4.1.1 xDB maintenance

If you are using xDB, you must:

- [Rebuild the xDB index](#).

- [Redeploy all the marketing definitions](#).

EXM configuration

EXM is enabled by default, to use it you must have an EXM license. For more information about configuring email delivery, see [Getting started with EXM for developers](#).

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 XP version where the changes were introduced.

You must perform all the steps that are applicable to 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 when the upgrade and installation are complete, you can delete your custom marketing definitions and marketing taxonomies again.

Note
If you try to remove any Sitecore XP items that are provided out of the box, they will not be removed. Since 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

2. At the bottom of the page, next to the name of the file that contains data about the restored items, click **Remove restored definition items**.
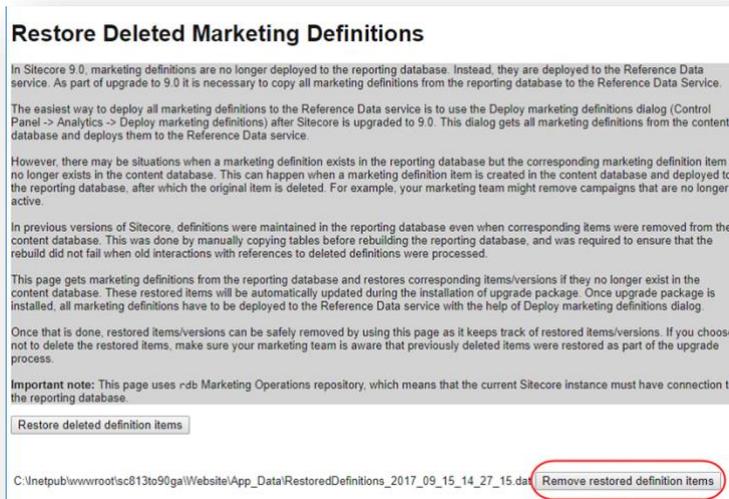
To delete unwanted marketing taxonomy items from the *Master* database:

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

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

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

Update customizations for reporting

In Sitecore XP, 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 XP 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.

Therefore you must:

- 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
```

2. In Sitecore XP, open the Path Analyzer admin page –
   `/sitecore/admin/pathanalyzer.aspx`.

3. 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 follow this step only 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:

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

2. Click **Start**.

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

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

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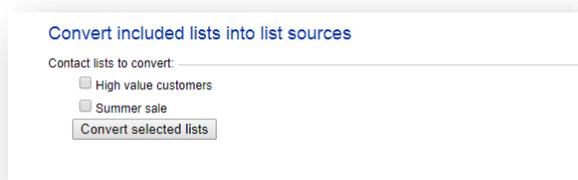
2. Select the lists that you want to convert into lists sources (only lists that contain included lists that are not treated as list sources appear on a screen) and click **Convert selected lists**

   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.

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

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 XP 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*.

Finish upgrading the Collection databases

Important
This section only applies if you are upgrading from Sitecore XP 9.0.0 - 9.2.0.

We recommend that you perform this step to align the database schema with the changes that were introduced in Sitecore XP 9.3.0.

You must perform the tasks described in this section if you are using SQL Azure and planning to use split/merge tool.

In SQL Azure, the aligned schema lets you use the scale-out feature with the help of the Microsoft Split/Merge service. For more information, see the article Split or merge xDB Collection database shards.

1. Ensure that all the *_Staging* tables are empty.

2. If the tables are not empty, delete all the data from the following tables:

```
DELETE FROM [xdb_collection].[CheckContacts_Staging]
DELETE FROM [xdb_collection].[ContactFacets_Staging]
DELETE FROM [xdb_collection].[ContactIdentifiers_Staging]
DELETE FROM [xdb_collection].[ContactIdentifiersIndex_Staging]
DELETE FROM [xdb_collection].[Contacts_Staging]
DELETE FROM [xdb_collection].[DeviceProfileFacets_Staging]
DELETE FROM [xdb_collection].[DeviceProfileFacets_Staging]
DELETE FROM [xdb_collection].[GetContactIdsByIdentifiers_Staging]
DELETE FROM [xdb_collection].[GetContactsByIdentifiers_Staging]
DELETE FROM [xdb_collection].[InteractionFacets_Staging]
DELETE FROM [xdb_collection].[Interactions_Staging]
DELETE FROM [xdb_collection].[UnlockContactIdentifiersIndex_Staging]
```

3. 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 missing percentile values*.

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

   If *Always Encrypted* (AE) is On:

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

   If AE is Off:

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

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

Since Sitecore XP 10.2.0, 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**.

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.

Chapter 5

Appendices

This chapter contains the following sections:

- Recommendations

- Appendix A – Scripts

- Appendix B – Generating shard keys and missing percentile values

- Appendix C - Update Experience Analytics custom segments

5.1 Recommendations

5.1.1 Enable telemetry reporting in production environments

The Sitecore telemetry reporting feature gathers information that helps Sitecore XP 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 that Sitecore XP 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:

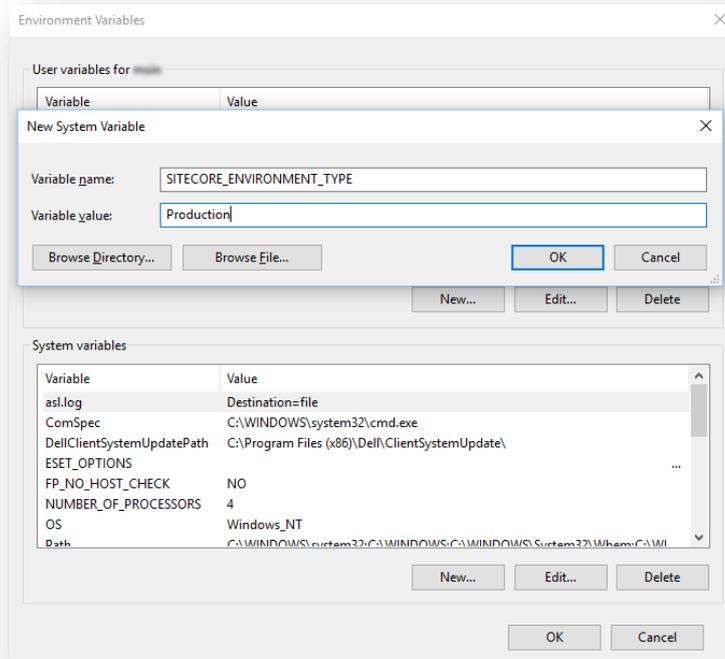1. In the Control Panel, click **System**, **Advanced system settings**.
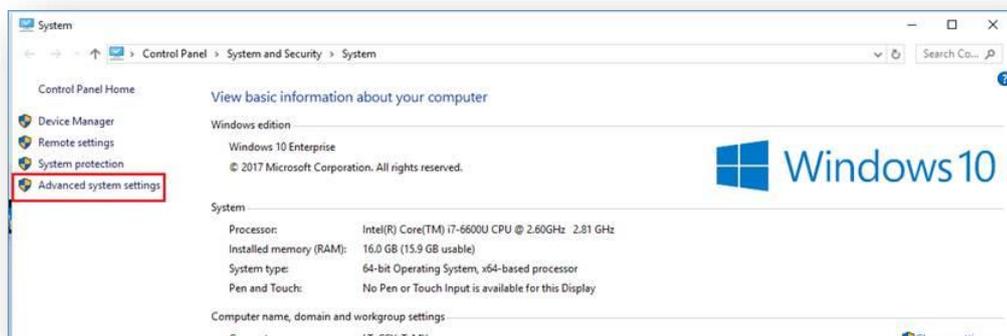
2. In the **System Properties** dialog, click **Environment Variables**.
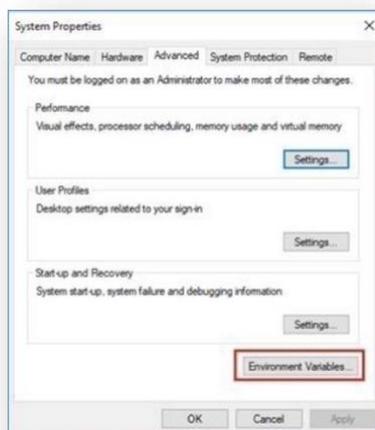
3. In the **Environment Variables** dialog, in the **System variables** section, click **New**.

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.

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 missing 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 from version 9.2.0 or earlier, the shard key values are null by default. Earlier upgrades to version 9.2.0 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 missing 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 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. 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 10.3.0 rev. 008463 (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
`--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= Xdb.Collection.Shard0" -c deviceProfiles

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

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

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

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

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

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.

5.5 Appendix D – Example of configuration patch

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
<?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`.

```
  </sitecore>
```