Sitecore Experience Platform Upgrade Guide

Sitecore Experience Platform 9.3.0

Upgrade to Sitecore XP 9.3.0 from Sitecore XP 8.1.0 - 9.0.1

Table of Contents

Chapter 1

Introduction

This guide describes how to upgrade from Sitecore Experience Platform 8.1.0 – 9.0.1 to Sitecore Experience Platform 9.3.0 and applies only to the on-premise version.

If you are upgrading from Sitecore XP 9.0.2 or later, use the *Upgrade to Sitecore XP 9.3.0 from Sitecore XP 9.0.2 or later guide* available at the Sitecore Downloads site.

If you are upgrading from Sitecore XP 8.1.0 – 9.0.1, you can upgrade directly to Sitecore XP 9.3.0. 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 9.3.0.

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

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

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

Chapter 2

Prerequisites

This chapter describes the prerequisites required to upgrade to Sitecore XP 9.3.0.

This chapter contains the following sections:

- Requirements
- Files and scripts for upgrading

2.1 Requirements

Before you upgrade to Sitecore XP 9.3.0, you should ensure that you meet the software and hardware requirements for this version of Sitecore XP. You must also download the upgrade files and prepare you environment for the upgrade process.

For more information about the requirements, see the Sitecore XP 9.3.0 Installation Guide.

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

2.1.1 Microsoft Visual C++ 2015 Redistributable Requirements

Sitecore XP 9.0.1 introduced a new prerequisite for the Microsoft Visual C++ 2015 Redistributable. For more information, see https://www.microsoft.com/en-us/download/details.aspx?id=53587.

Note
This redistributable 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.*

2.1.2 Obtain and set up certificates for SSL

Note
If you already use HTTPS, skip this section.

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

For more information about how to set up SSL in IIS, see Microsoft's documentation.

For more information about how to install the server certificates, see the Installation Guide.

2.2 Files and scripts for upgrading

This section lists the files required to upgrade Sitecore XP. Download the files from the Sitecore Experience Platform download site.

To upgrade from Sitecore XP 8.1 – 9.0.1, you must have:

- One of the following files depending on which version of Sitecore XP you are upgrading from:
  - `Sitecore 9.3.0 rev. 003498 (update package for 8.1).zip`
  - `Sitecore 9.3.0 rev. 003498 (update package for 8.2).zip`
  - `Sitecore 9.3.0 rev. 003498 (update package for 9.x).zip`

  Each of these files contains the appropriate update package for that version of Sitecore, for example:
  - `Sitecore 9.3.0 rev. 003498 (update package for 9.x).update`

  Unpack the appropriate version of the update package and save it in a suitable location.
- `Sitecore Update Installation Wizard 4.0.1 rev.00153 for 8.x-9.0.zip`
- `Sitecore 9.3.0. rev. 003498 (upgrade files).zip`

  On the Download site, this file is referred to as *Configuration files for upgrade* and contains:
  - `Database Upgrade Script.zip`

    This file contains several files, including the database upgrade scripts. The upgrade scripts contain changes that must be applied to the Sitecore databases across every version of Sitecore to ensure compatibility. They also modify certain tables and collections to support new functionality and improve performance. The changes must be applied once per database.
  - `Databases.zip`

    This file contains the new databases that you deploy during the upgrade process.
  - `Disable xDB.zip`

    This file contains the patch files that help you turn off xDB during the upgrade process.
  - `MarketingDefinitionsUpgrade.zip`

    This file contains the `.aspx` pages that help you prepare your marketing definitions for the upgrade.
- xDB Data Migration Tool – required if you have existing analytics data in MongoDB that you want to use with Sitecore XP 9.3.0.

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

Chapter 3

Prepare for the Upgrade

Before you upgrade to Sitecore XP 9.3.0, you must complete preparatory tasks.

This chapter describes the preparatory tasks for each upgrade path and includes a section describing the tasks that you must perform for all the upgrade paths.

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

This chapter contains the following sections:

- Disable the Sitecore modules
- To upgrade from Sitecore XP 8.1 – Sitecore XP 8.2 Update 7
- To upgrade from Sitecore XP 9.0.0 – 9.0.1
- Final preparations for all versions

3.1 Disable the Sitecore modules

You must disable any Sitecore modules that you are using.

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

3.1.1 Web Forms for Marketers

Sitecore XP 9.3.0 does not support the Web Forms for Marketers module. This module was deprecated in Sitecore XP 9.1.0.

If you are using the Web Forms for Marketers module, you must disable it before you upgrade to Sitecore XP 9.3.0.

To disable the Web Forms for Marketers module:

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

3.1.2 Email Experience Manager

If you upgrade from Sitecore XP 8.x and used Email Experience Manager module, you must disable it before starting the upgrade process:

To disable EXM:

1. Delete the following folders:
   - `\App_Config\Include\EmailExperience`
   - `\App_Config\Include\Z.EmailExperience`
2. Delete the following file:
   - `\App_Config\Include\ExperienceProfile\Sitecore.ExperienceProfile.Reporting.Sitecore.EmailExperience.ExperienceProfile.config`
3. In the `\bin` folder of your website, delete all the files that begin with `Sitecore.EmailCampaign`, `Sitecore.EDS`, or `Sitecore.ExM`.

3.1.3 Sitecore Experience Accelerator

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

3.1.4 Data Migration Tool

If you used the Data Migration Tool 2.0 to migrate your data to Sitecore XP 9.0, you must uninstall the tool before upgrading to Sitecore XP 9.3.0. After you have migrated your data, you no longer need the tool.

For more information about how to uninstall the Data Migration Tool, see the Data Migration Tool Installation Guide.

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

3.1.5 Disable active Sitecore services

Before upgrading, you must disable some active services.

To disable the active services:

1. Delete all the `*.JavaScriptServices.*` files from the `bin` and `App_config` folders and any child folders.
2. In the `App_Config\Sitecore\LayoutService\` folder, delete the `Sitecore.LayoutService.Jss.config` file.
3. In the `App_Config\Sitecore\Services.GraphQL\` folder, delete the `zSitecore.JavaScriptServices.GraphQL.config` file.

3.1.6 Disable custom DLLs and configuration files

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

3.2 To upgrade from Sitecore XP 8.1 – Sitecore XP 8.2 Update 7

Important
If you are upgrading from Sitecore XP 8.1.0 – Sitecore XP 8.2 Update 7 to Sitecore XP 9.3.0, review the steps in this section.

If you are upgrading from Sitecore XP 9.0.0 – 9.0.1 to Sitecore XP 9.3.0, skip this section.

3.2.1 Restore deleted marketing taxonomies and marketing definitions

If you are using the Sitecore Experience Database (xDB) functionality, (the `Xdb.enabled` setting is set to `true`, and xDB is enabled in the license), before you install the upgrade package, you must restore any of the standard marketing taxonomies and marketing definitions that you have deleted from your Sitecore installation.

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

Before you restore anything:

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

To restore deleted marketing taxonomy items:

1. Open the *Restore Deleted Marketing Taxonomies* page by entering the following URL in a browser:

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

   Important
   This page requires direct access to the *Reporting* database. Ensure that the current Sitecore instance has the appropriate connection string in the `\App_Config\ConnectionStrings.config` file and that the database is accessible.

   You can access this page from a CM instance.

2. Click *Restore deleted taxonomy items*.

   All the default marketing taxonomy items are restored to the *Master* database. A `RestoredTaxonomies_{timestamp}.dat` file is created in the `\App_Data` folder. This file lists all the restored items. When the upgrade is completed, if you want to delete these unwanted marketing taxonomies again, use this file to identify them.

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

Important
This page uses the Marketing Operations repository in the *Reporting* database. The current Sitecore instance must therefore have access to the *Reporting* database.

2. Click **Restore deleted definition items**.

   All the default marketing definition items are restored from the *Reporting* database to the *Master* database. A `RestoredDefinitions_{timestamp}.dat` file is created in the `\App_Data` folder. This file lists all the restored items. When the upgrade is completed, if you want to remove these marketing definitions again, use this file to identify them.

3.2.2 Validate the names of the marketing definition items

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

To validate the names of the marketing definition items:

1. Unpack the `MarketingDefinitionsUpgrade.zip` file.
2. Copy the `DefinitionItemsNameValidator.aspx` page to the `\sitecore\admin` folder of your website.
3. On a CM instance, open the *Definition Items Name Validator* page by entering the following URL in a browser:

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

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

   The name of the marketing definition item is used as the value of the *Alias* property. The page validates the names of the marketing definition items so that the names can be fixed before the update package is installed.

   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.
4. Click **Validate**.

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

   This report lists all the validation errors that occurred.
5. Ensure you manually fix all the issues that were found, and then validate the names again.

   Repeat this procedure until all the names have been validated.

3.2.3 Upgrade the databases

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

To upgrade the databases:

1. Extract the database upgrade scripts from the `Database Upgrade Script.zip` file that you downloaded earlier.
2. Run the `CMS_core_master_web8x.sql` script for the *Core*, *Master*, and *Web* databases.
3. Run the `CMS_core.sql` script for the *Core* database only.

3.2.4 Deploy the new databases

In Sitecore XP 9.0, the format of all the databases changed and they are now distributed as *Data-tier Application Component Packages (DACPAC)*. You can find the databases that must be updated as part of this upgrade in the `Sitecore 9.3.0 rev. 003498 (upgrade files).zip` package, in the *Databases.zip* package.

To use the *Sitecore Forms* component and the Sitecore Experience Database (xDB) functionality, you must deploy the new databases.

When you deploy a new database, you must also create a user and assign the appropriate access rights. For more information about how to specify user credentials when you deploy a database, see *Appendix A: Scripts*.

Sitecore Forms

To use the *Sitecore Forms* component:

1. Deploy the `Sitecore.Experienceforms.dacpac` to SQL Server.
2. In the `\App_Config` folder, in the `ConnectionStrings.config` file, add the connection string for the `Sitecore.ExperienceForms` database. For example:

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

The `Experience Forms` connection string is required for the *content management* and *content delivery* server roles.

xDB

To use the xDB functionality:

1. Deploy the `Sitecore.Processing.tasks.dacpac` to SQL Server.
2. In the `\App_Config` folder, in the `ConnectionStrings.config` file, add the connection string for the `Xdb.Processing.Tasks` database. For example:

```
<add name="xdb.processing.tasks" connectionString="user id=user;password=password;Data
Source=(server);Database=Sitecore.Processing.Tasks" />
```

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

3.2.5 Enable Email Experience Manager

Note
You must manually migrate your EXM (Email Experience Manager) data to Sitecore XP 9.3.0 – it is not migrated during the upgrade.

For more information about migrating EXM data, see the Sitecore EXM Data Migration Guide.

To enable EXM:

1. Deploy the following files to SQL server:
   - `Sitecore.Exm.master.dacpac`
   - `Sitecore.Messaging.dacpac`
2. You must create SQL logins with the appropriate access rights to the *Exm.master* and *Messaging* databases. These SQL logins must be members of the *db_datareader* and *db_datawriter* security roles.
3. In the `\App_Config` folder, in the `ConnectionStrings.config` file, add the connection strings for the *Exm.master* and the *Messaging* databases.

   For example:

   ```
   <add name="exm.master" connectionString="user id=exmmasteruser;password=Test12345;Data
   Source=(server);Database=xp0_Exm.master" />
   <add name="messaging" connectionString="user id=messaginguser;password=Test12345;Data
   Source=(server);Database=xp0_Messaging" />
   ```

   The `exm.master` connection string is only required on every *Content Management* server.

   The `messaging` connection string is required on every *Content Management* and *Content Delivery* server.
4. Add the following three connection strings:
   - `EXM.InternalApiKey`

     You only need to add this connection string if you want to add a dedicated email dispatch server.
   - `EXM.CryptographicKey`
   - `EXM.AuthenticationKey`

   The keys must be represented in hexadecimal format by 64 characters and you can only use the symbols 0-9 and A-F.

   For example:

   ```
   <add name="EXM.CryptographicKey" connectionString=
   "E040C938FC9E4EBC3E93330B0F7837F284207B8180DB64CB5B6ABEB1AFBF6F5B" />
   ```

3.3 To upgrade from Sitecore XP 9.0.0 – 9.0.1

Important
If you are upgrading from Sitecore XP 9.0.0 – 9.0.1 to Sitecore XP 9.3.0, review the steps in this section.

If you are upgrading from Sitecore XP 8.1.0 – Sitecore XP 8.2 Update 7 to Sitecore XP 9.3.0, skip this section.

3.3.1 Upgrade the databases

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

To upgrade the databases:

1. Extract the database upgrade scripts from the `Database Upgrade Script.zip` file that you downloaded earlier
2. Run the `CMS_core_master_web9x.sql` script for the *Core, Master,* and *Web* databases.
3. Run the `CMS_core.sql` script for the *Core* database'
4. Run the `SXP_processing_tasks.sql` script for the *Processing.tasks* database.
5. Run the `SXP_experienceforms_storage.sql` for the *ExperienceForms* database.
6. Run the `SXP_experienceforms_filestorage.sql` for the *ExperienceForms* database.

3.3.2 Disable Experience Analytics

If you are using the xDB functionality, complete this procedure.

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

1. In the `Sitecore 9.2.0 rev. 002893 (upgrade files).zip` file, locate the `Disable xDB.zip` file and unpack it to find the `Disable.xAnalytics.config` file.
2. In the `\App_Config\Include` folder, create a `\Z.Custom` folder and copy the `Disable.xAnalytics.config` file to the `\App_Config\Include\Z.Custom` folder.

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

3.3.3 Stop the Windows services

If you are using the xDB functionality, complete this procedure.

Stop the following Windows services:

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

3.3.4 Enable Email Experience Manager (EXM)

This section contains information about how to upgrade EXM from:

- Sitecore XP 9.0 Initial Release
- Sitecore XP 9.0.1

Upgrade from Sitecore XP 9.0 Initial Release

Note
If you are upgrading from Sitecore XP 9.0.1, skip this section.

To enable EXM:

1. Deploy the following files to SQL Server:
   - `Sitecore.Exm.master.dacpac`
   - `Sitecore.Messaging.dacpac`
2. Create SQL logins with the appropriate access rights to the *Exm.master* and *Messaging* databases. These SQL logins must be members of the *db_datareader* and *db_datawriter* security roles.

   When you deploy a new database, you must also create a user and assign the appropriate access rights. For more information about how to specify user credentials when you deploy a database, see *Appendix A: Scripts*.
3. In the `\App_Config` folder, in the `ConnectionStrings.config` file, add connection strings for the *Exm.master* and the *Messaging* databases.

   For example:

   ```
   <add name="exm.master" connectionString="user id=exmmasteruser;password=Test12345;Data
   Source=(server);Database=xp0_Exm.master" />
   <add name="messaging" connectionString="user id=messaginguser;password=Test12345;Data
   Source=(server);Database=xp0_Messaging" />
   ```

   The `exm.master` connection string is only required on every *Content Management* server.

   The `messaging` connection string is required on every *Content Management* and *Content Delivery* server.
4. Add the following three connection strings:
   - `EXM.InternalApiKey`

     You only need to add this connection string if you want to add a dedicated email dispatch server.
   - `EXM.CryptographicKey`
   - `EXM.AuthenticationKey`

   The keys must be represented in hexadecimal format by 64 characters and you can only use the symbols 0-9 and A-F.

   For example:

   ```
   <add name="EXM.CryptographicKey" connectionString=
   "E040C938FC9E4EBC3E93330B0F7837F284207B8180DB64CB5B6ABEB1AFBF6F5B" />
   ```

Upgrade the *EXM.Master* database from Sitecore XP 9.0.1

If you are upgrading from Sitecore XP 9.0.1, run the following scripts for the *EXM.Master* database:

- `SXP_exm_master_dispatchqueue.sql`
- `SXP_exm_master_suppressions.sql`
- `SXP_exm_master_save_suppressions.sql`
- `SXP_exm_master_update_suppressions.sql`

3.3.5 Indexing and performance

Indexing is performed during the upgrade. Depending on the version of Sitecore you are upgrading from, the upgrade can take a long time.

To speed up the upgrade process, disable indexing by removing the following processor from the `Sitecore.ContentSearch.config` file:

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

3.4 Final preparations for all versions

3.4.1 Add new connection strings

In Sitecore XP 9.0.2, we added a new *PackageManagementServiceUrl* connection string that enables you to use the Update Center to upgrade your Sitecore installation. This feature is not supported when you upgrade from Sitecore XP 8.1-9.0.1 to Sitecore XP 9.3.0. You will be able to use it for your next upgrade.

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

Before you upgrade to Sitecore XP 9.3.0, you must add the *security* and *package management* connection strings.

To add the connection strings:

1. Open the `\App_Config\ConnectionStrings.config` file.
2. Add the *security* connection string, for example:

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

3. Add the *PackageManagementServiceUrl* connection string:

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

3.4.2 Stop all content tests and disable xDB

If you are using the Sitecore Experience Database (xDB) functionality, you must perform these steps.

Stop active content tests

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

Disable xDB

To disable xDB:

1. In the `Sitecore 9.3.0 rev. 003498 (upgrade files).zip` file that you unpacked earlier, locate the `Disable xDB.zip` file.
2. Create a `\App_Config\Include\Z.Custom` folder and copy the `Disable.xDB.config` file to this folder.

Important
When the upgrade is finished, you must re-enable xDB by removing the `Disable.xDB.config` file from the `\App_Config\Include\Z.Custom` folder.

3.4.3 Upgrade the reporting database

Note
If you are going to use the xDB Data Migration Tool to migrate your xDB data, the tool will upgrade the database and you can skip this section.

If you are using the xDB functionality, you must upgrade the *Reporting* database.

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

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

   If you want rates other than 0 percent, do one of the following:

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

     For conversion rates:

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

     For bounce rates:

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

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

2. 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.

Chapter 4

Upgrade the Sitecore xConnect Server

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

This chapter describes how to upgrade xConnect from:

- Sitecore XP 9.0.0
- Sitecore XP 9.0.1

Sitecore XP 9.1.0 introduced the new Sitecore Cortex™ Processing Engine role and you can use with or without Microsoft Machine Learning Server (MLS). If you want to use the it in Sitecore XP 9.3.0, you must install and configure it. This chapter describes how to install it.

For more information about installing the Sitecore Cortex Processing Engine, including the deployment topologies, see the Sitecore Experience Platform 9.3.0 Installation Guide.

This chapter contains the following sections:

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

4.1 Prepare to upgrade the xConnect Server

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

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

4.1.1 Prerequisites for upgrading xConnect

Before you upgrade xConnect:

1. Download the following files:
   - `Sitecore 9.3.0 rev. 003498 (WDP XP0 packages).zip`.

     In the *Download options for On Premises deployment* section, this file is called *Packages for XP Single*.
   - `Sitecore 9.3.0 rev. 003498 (upgrade files).zip`.

     In the *Upgrade options* section, this file is called *Configuration files for upgrade*. You might have already downloaded this file if you followed the procedures in the section *Files and scripts for upgrading*.
2. Unpack the `Sitecore 9.3.0 rev. 003498 (WDP XP0 packages).zip` file and then unpack the `Sitecore 9.3.0 rev. 003498 (OnPrem)_xp0xconnect.scwdp.zip` file.
3. Unpack the `Sitecore 9.3.0 rev. 003498 (upgrade files).zip` file and then unpack the `Database Upgrade Script.zip` file. This file contains several database upgrade scripts.

   You might have already unpacked this file if you followed the steps in the section *Files and scripts for upgrading*.
4. If you have any functionality that accesses the contact/interaction data extraction process from outside of xConnect, you must stop it. You can restart it when the upgrade is completed. An example of this kind of functionality is rebuilding the *Reporting* database.

   Important
   If you are upgrading from Sitecore XP 9.0.1 or earlier, you cannot pause these tasks before the upgrade and resume them afterwards because the extraction cursors were changed in Sitecore XP 9.1.1.

4.1.2 Backup your customizations

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

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

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

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

4.1.3 Stop Windows services

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

Stop the following Windows services:

- Sitecore XConnect Search Indexer
- Sitecore Marketing Automation Engine

4.2 Upgrade the xDB databases

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

4.2.1 Messaging database

Deploy the Messaging database

If you are upgrading from Sitecore XP 9.0 Initial Release or earlier, you must deploy the *Messaging* database. For more information about deploying the *Messaging* database, see the section *Enable Email Experience Manager*.

After you have deployed the *Messaging* database, in the following files:

- `\xconnect\App_Config\ConnectionStrings.config`
- `\xconnect\App_data\jobs\continuous\AutomationEngine\App_Config\ConnectionStrings.config`

  Add the *messaging* connection string, for example:

```
<add name="messaging" connectionString="user id=messaginguser;password=Test12345;data source=source; Database=Sitecore.Messaging" />
```

Note
You must add the *messaging* connection string on every xConnect server that performs the *Sitecore Marketing Automation Engine* role.

Upgrade the Messaging database

If you are upgrading from Sitecore XP 9.0.1, you must upgrade the *Messaging* database.

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

4.2.2 xDB Reference Data database

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

To upgrade the *xDB Reference Data* database:

1. Run the `SXP_referencedata_Part1.sql` script.
2. If you receive the following message:

   # definitions cannot be migrated. Length of # definition monikers exceeds 300 characters. Consider truncate, change or remove them. Upgrade stopped. Fix all the issues and run this script again. Upgrade (part 1) has not been completed.

   Fix all the issues as described in this message and run the `SXP_referencedata_Part1.sql` script again.

   Repeat this step until you receive the *"definitions cannot be migrated"* message.
3. Run the `SXP_referencedata_Part2.sql` script.

4.2.3 xDB Collection database on SQL Server

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

Upgrade the *Collection* database - Shard Map Manager

If you are going to apply scaling to the *xDB Collection* database, follow this procedure.

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

Upgrade the xDB Collection database – Shards

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

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

  You must specify the user name in the script.

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

With Always Encrypted

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

1. Run the `SXP_collection_AE_Part1.sql` script for every shard.
2. In the new *[UnlockContactIdentifiersIndex_Staging]* table, configure deterministic encryption for the *[Identifier]* and *[Source]* columns, for every shard.

   To simplify the configuration process, SQL Server Management Studio provides an Encryption Wizard.
3. Run the `SXP_collection_AE_Part2.sql` script for every shard.

Without Always Encrypted

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

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

SQL Azure

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

To use the split/merge service:

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

   For more information, see *Appendix B: Generating shard keys and missed percentile values*.
2. After you have run the *Sitecore Xdb Collection Update* tool, run the `SXP_collection_Part2.sql` script for every shard.

4.2.4 Upgrade the Collection database on MongoDB

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

If you are going to move from SQL Server 2016 to MongoDB as part of the upgrade, skip this section.

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

To continue using the *xDB Collection* database on the MongoDB provider, upgrade your MongoDB instance to version 4.0.5 – 4.0.10.

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

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

To build indexes efficiently, follow the suggestions in the article Index Build Operations on a Populated Collection.

4.2.5 Other databases

To upgrade the other databases:

1. Run the `SXP_processing_pools.sql` script for the *Xdb Processing Pools* database.
2. Run the `SXP_marketingautomation.sql` script for the *Xdb Marketing Automation* database.

4.3 Update the files

Update the files on the xConnect instance.

Note
You can use a Windows batch script to automate steps 1-2 in the following procedure. For more information about the batch script, see *Appendix A: Scripts*, *xConnect application preparation script.*

To update the files:

1. Delete all the files under the following folders, but leave the folders themselves to ensure that the file permissions are preserved:
   - `\Bin`
   - `\App_Data\collectiondeployment`
   - `\App_Data\Config`
   - `\App_Data\Copyrights`
   - `\App_Data\Diagnostics`
   - `\App_Data\Models`
   - `\App_Data\solrcommands`
   - `\App_Data\jobs\continuous\AutomationEngine`
   - `\App_Data\jobs\continuous\AutomationEngine\App_Data\Config`
   - `\App_Data\jobs\continuous\AutomationEngine\App_Data\Diagnostics`
   - `\App_Data\jobs\continuous\IndexWorker`
   - `\App_Data\jobs\continuous\IndexWorker\Copyrights`
   - `\App_Data\jobs\continuous\IndexWorker\App_Data\Config`
   - `\App_Data\jobs\continuous\IndexWorker\App_Data\Diagnostics`
   - `\App_Data\jobs\continuous\IndexWorker\App_Data\Models`
2. In the `\App_Data\jobs\continuous\IndexWorker` folder, delete the following language folders if they exist:
   - `\de-DE`
   - `\es-ES`
   - `\fr-FR`
   - `\it-IT`
   - `\ja-JP`
   - `\ko-KR`
   - `\pt-BR`
   - `\ru-RU`
   - `\zh-CN`
   - `\zh-TW`
3. Copy all the files and folders from the `Sitecore 9.3.0 rev. 003498 (OnPrem)_xp0xconnect.scwdp\Content\Website` folder that you unpacked earlier, except for the following folders:
   - `\App_Config`
   - `\App_Data\jobs\continuous\AutomationEngine\App_Config`

- `\App_Data\jobs\continuous\IndexWorker\App_Config`
- `\App_Data\logs`

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

Note
When you copy the `App_Data\jobs\continuous\ProcessingEngine` folder, you get all the files for the processing engine.

4.3.1 Add new Connection strings

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

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

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

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

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

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

4.3.2 Reconfigure xConnect Search Indexer service

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

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

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

4.3.3 Reconfigure MA Engine service

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

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

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

4.3.4 Restore customizations

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

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

4.4 Install and configure the Sitecore Cortex Processing Engine

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

To deploy the databases, in the `Sitecore 9.3.0 rev. 003498 (OnPrem)_xp0xconnect.scwdp\Content` folder, find and deploy the following files to SQL server:

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

To configure the Processing Engine:

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

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

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

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

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

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

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

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

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

4.4.1 Install and configure Microsoft Machine Learning Server

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

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

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

Chapter 5

Install xConnect

If you want to use the Sitecore Experience Database (xDB) functionality in Sitecore XP 9.3.0, you must install and configure Sitecore xConnect.

Important
If you are upgrading from Sitecore XP 8.1.0 – Sitecore XP 8.2 Update 7, you must install and configure Sitecore xConnect.

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

If you are upgrading from Sitecore XP 9.0 – Sitecore XP 9.0.1 *without* xConnect installed and want to use the Sitecore Experience Database (xDB) functionality in Sitecore XP 9.3.0, you must perform the tasks described in this chapter.

This chapter contains the following section:

- Install and configure xConnect

5.1 Install and configure xConnect

For more detailed information about which topologies support xConnect, see the Sitecore Experience Platform 9.3.0 Installation Guide.

To configure Sitecore xConnect for a single instance Sitecore solution:

1. Install xConnect.

   For more information, see the *Sitecore Experience Platform 9.3.0 Installation Guide*.

2. For a single instance Sitecore solution that you have upgraded, in the `\App_Config` folder, in the `ConnectionStrings.config` file, add the following connection strings and modify all the parameters:

```
  <add name="xconnect.collection" connectionString="https://Sitecore_xconnect"/>
  <add name="xconnect.collection.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;Fi
ndValue="/>
  <add name="xdb.referencedata" connectionString="user id=user;password=password;Data
Source=(server);Database=Sitecore_ReferenceData"/>
  <add name="xdb.referencedata.client" connectionString="https://Sitecore_xconnect"/>
  <add name="xdb.referencedata.client.certificate"
connectionString="StoreName=My;StoreLocation=
LocalMachine;FindType=FindByThumbprint;FindValue="/>
  <add name="xdb.processing.pools" connectionString="user
id=user;password=password;Data Source=(server);Database=Sitecore_Processing.Pools"/>
  <add name="xdb.marketingautomation" connectionString="user
id=user;password=password;Data
Source=(server);Database=Sitecore_MarketingAutomation"/>
  <add name="xdb.marketingautomation.reporting.client"
connectionString="https://Sitecore_xconnect"/>
  <add name="xdb.marketingautomation.reporting.client.certificate"
connectionString="StoreName=My;StoreLocation=
LocalMachine;FindType=FindByThumbprint;FindValue="/>
  <add name="xdb.marketingautomation.operations.client"
connectionString="https://Sitecore_xconnect"/>
  <add name="xdb.marketingautomation.operations.client.certificate"
 connectionString="StoreName=My;StoreLocation=
LocalMachine;FindType=FindByThumbprint;FindValue="/>
  <add name="messaging" connectionString="user id=user;password=password;Data
Source=(server);Database=Sitecore.Messaging"/>
  <add name="sitecore.reporting.client" connectionString="https://sc910 xconnect" />
  <add name="sitecore.reporting.client.certificate"
connectionString="StoreName=My;StoreLocation=LocalMachine;FindType=FindByThumbprint;Fi
ndValue=C7C27AAE1E197AC0B743BDACF18E21C214C9CAB5" />
```

Note
These connection strings are for a single instance Sitecore solution only. For a scaled environment, see the Sitecore server role configuration reference documentation.

Grant permissions to the xConnect certificates

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

To grant the read permissions:

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

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

5. Restart IIS.

Chapter 6

Install the Upgrade package

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

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

This chapter contains the following sections:

- Upload and analyze the upgrade package
- The upgrade package analysis report
- Install the upgrade package

6.1 Upload and analyze the upgrade package

To install the upgrade package:

1. On the Sitecore Launchpad, click **Control Panel**, and in the **Administration** section, click **Install a package**.
2. Install the `Sitecore Update Installation Wizard 4.0.1 rev. 00153 for 8.x-9.0.zip` package.

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

3. On the Sitecore Launchpad, click **Control Panel**, and in the **Administration** section, click **Install an update**.

   Alternatively, you can open the **Update Installation Wizard** by entering the following URL in your web browser:

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

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

4. On the **Welcome to Sitecore update installation wizard** page, click **Select a package**.
5. On the **Select a package** page, click **Choose File** and navigate to the folder where you saved the update file for Sitecore XP 9.3.0.
6. Select the update file for the Sitecore version that you are upgrading from, and then click **Open**.

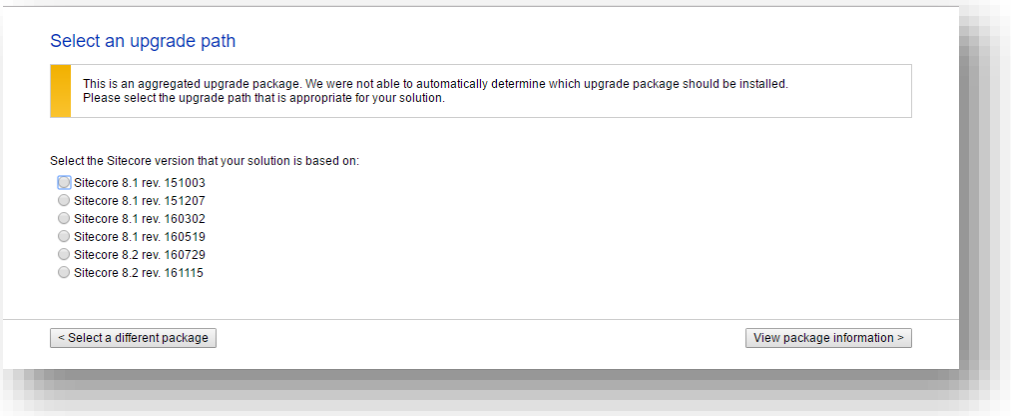
   The update file uses the following naming pattern:

   `Sitecore 9.3.0 rev. 003498 (update package for 9.x).update`

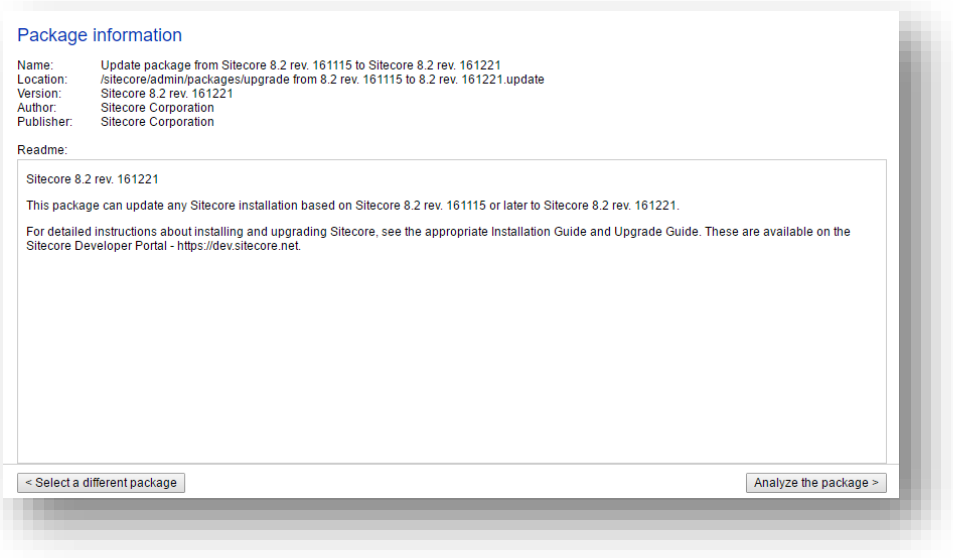
   For more information about the update file, see the section *Files and scripts for upgrading*.

7. Click **Package information**.

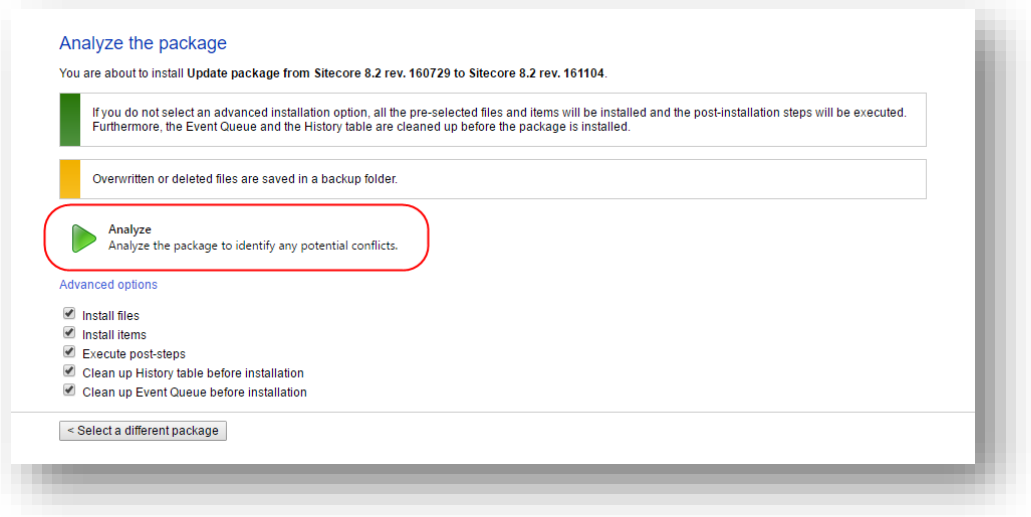
   If the **Update Installation Wizard** cannot detect your current Sitecore version automatically, on the **Select an upgrade path** page, you must manually select the Sitecore version and revision number that your solution is based on, and then click **View package information**.



8. On the **Package information** page, review the information about the package, and then click **Analyze the package**.



9. On the **Analyze the package** page, click **Advanced options**, review the installation options**.**



   In the **Advanced options** section, you must select the **Execute post-steps** check box, regardless of the other check boxes you might have selected.

   The post-steps are performed at the end of the upgrade process. These steps can include items or file operations that are executed by the Sitecore API and can vary according to the Sitecore version that you are upgrading from.

   During the analysis, the installation wizard identifies any potential conflicts in the configuration files being upgraded and any breaking changes in custom code.

10. Click **Analyze**.
11. After the analysis process is complete, click **Analysis result**.

6.2 The upgrade package analysis report

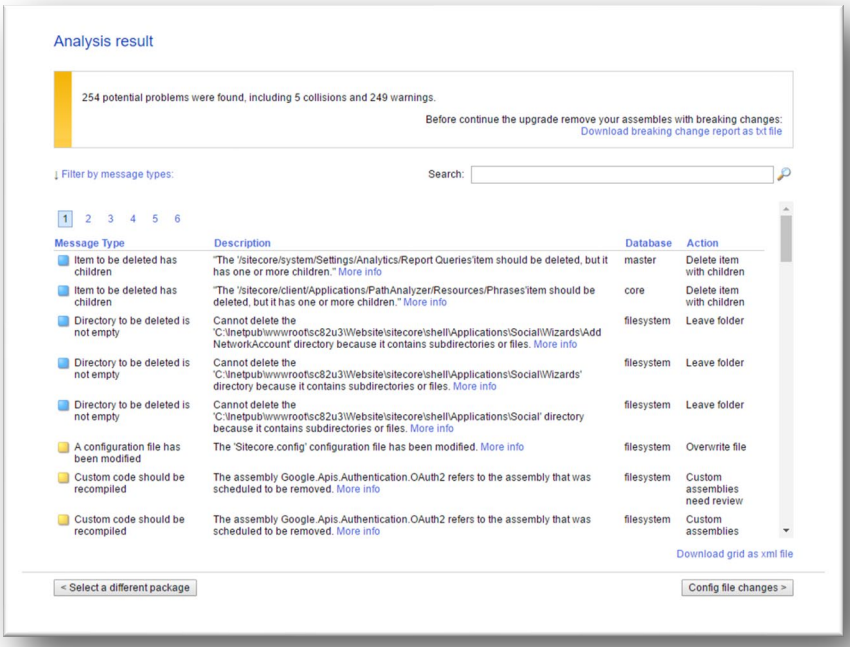
The analysis phase detects any problems that can occur during the upgrade. The analysis report contains details about collisions, errors, and warnings that you must deal with before performing the upgrade.

If the **Update Installation Wizard** does not find any conflicts, you can proceed to the section *Install the upgrade package*.

Collisions, errors, and warnings can occur when there is a risk of losing data. For example, you can lose data when the upgrade involves deleting an item that has children or a folder that contains files.

In the analysis report, collisions are marked with a blue square and warnings with a yellow square. However, if the wizard finds errors, it will display a red square.

The action that the wizard will take to resolve each collision or warning is listed in the **Action** column.



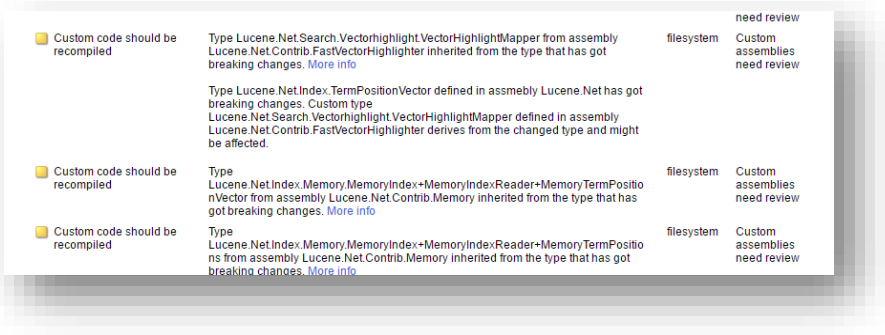
We recommend that you review all the collisions, errors, and warnings, as well as the actions that will be performed to resolve them.

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

6.2.1 Resolve breaking changes in custom code

If the wizard detects some custom code that will be affected by breaking changes that were introduced in a Sitecore version, the **Analysis result** page displays a warning.

The warning comes with an action and a description. To see more information, click **More info**.



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

To resolve these warnings:

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

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

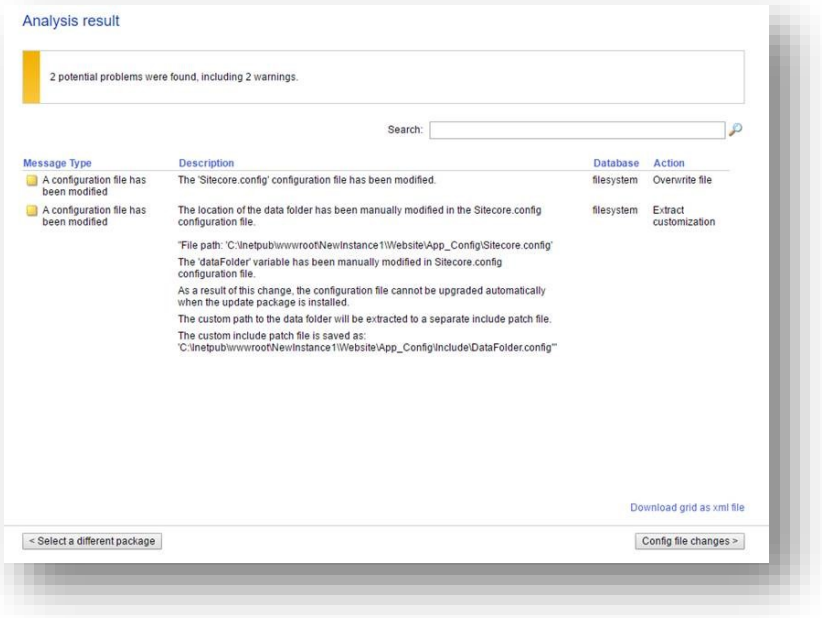
  Or

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

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

6.2.2 Resolve configuration file conflicts

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

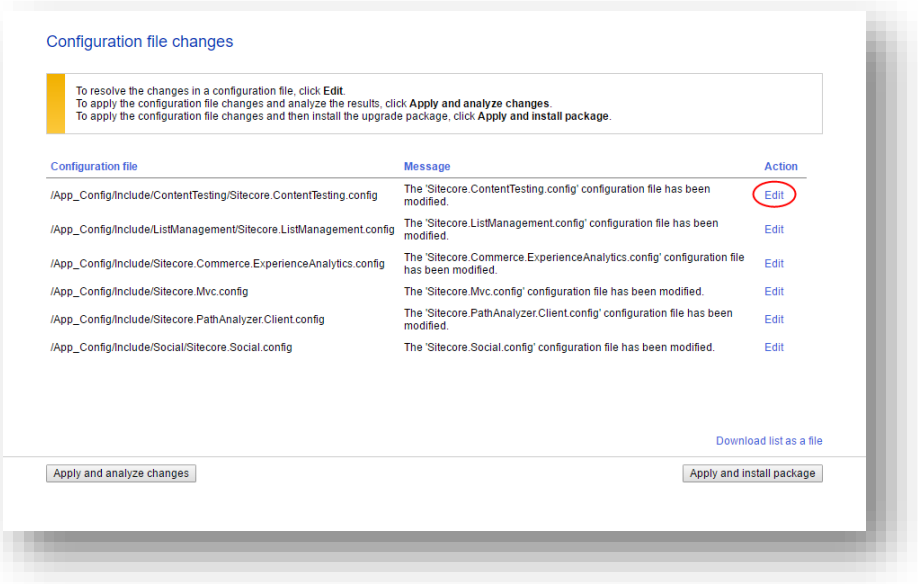


Some of these configuration file conflicts can be resolved automatically and others must be resolved manually.

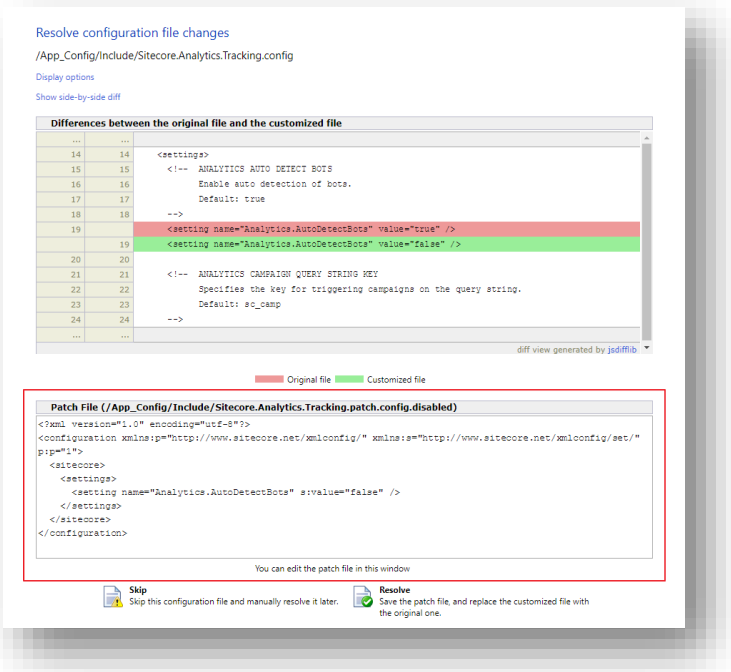
To see more information about a configuration file change, click the description.

To resolve configuration file conflicts:

1. On the **Analysis result** page, click **Config file changes**.



2. To automatically resolve all the configuration file conflicts and install the package, click **Apply and install package**. When you are finished, skip to step 5.
3. To manually resolve a configuration file conflict, in the **Action** column for the configuration file, click **Edit**.
4. On the **Resolve configuration file changes** page, review the differences between the original file and the customized file.



5. To manually resolve the conflicts, in the **Patch File** panel, make the necessary changes, and then click **Resolve**.
6. To skip a conflict and manually resolve it later, click **Skip**.

   You must manually review your configuration patch files after the upgrade is complete.

   Note
   When you skip the file, the **Update Installation Wizard** automatically detects any customizations in the Sitecore configuration files, creates the patch, and replaces the customized file with the new file when it installs the upgrade package.

7. After you have dealt with all the configuration file conflicts, on the **Configuration file changes** page, click **Apply and analyze changes**.

   At this point, the **Update Installation Wizard** restarts the analysis and ensures that all the configuration file conflicts have either been resolved or skipped.

   The **Update Installation Wizard** then creates new patch files with the customizations and disables them by adding the `.disabled` extension to the files.

   Note
   In certain scenarios, the generated patch files can contain customizations that are critical for the Sitecore instance to run correctly. To avoid breaking changes, the **Update Installation Wizard** displays an error on the **Apply and analyze changes** page.

   If this happens, you must enable the patch files by removing the `.disabled` extension, and then refresh the **Update Installation Wizard** page in your browser.

   The patch files are created and stored in the same folder as the original configuration files.

8. When the analysis is complete, click **Analysis result**.

   Note
   If the analysis fails, you must restart the **Update Installation Wizard**. If this does not solve the problem, contact Sitecore support.

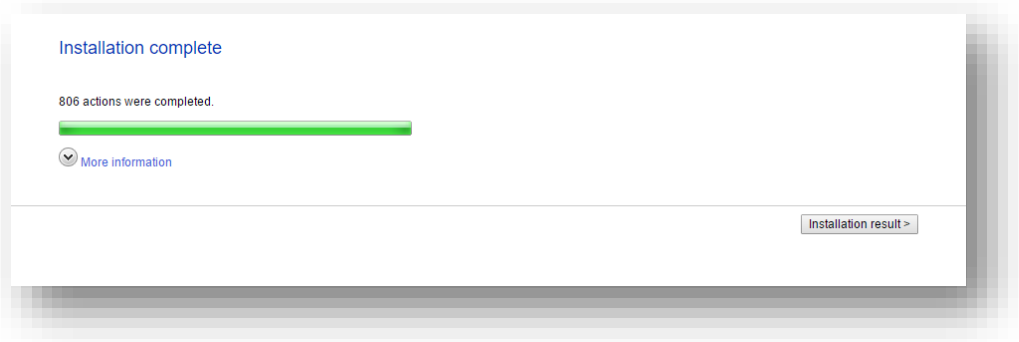
6.3 Install the upgrade package

If the **Update Installation Wizard** does not find any configuration file conflicts, or you have resolved all of them, you can install the upgrade package.

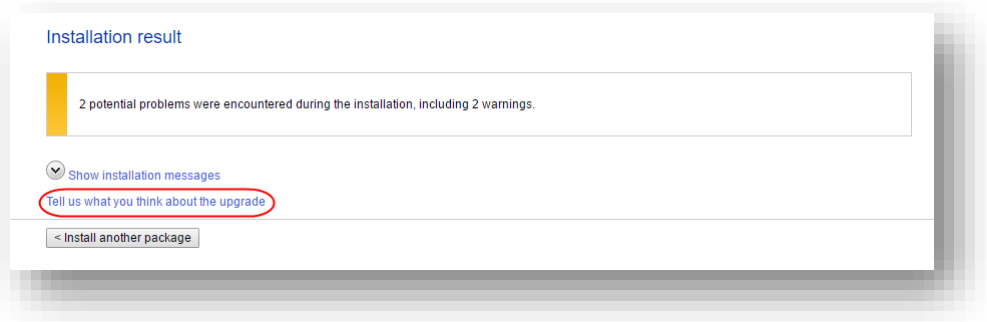
1. In the **Update Installation Wizard**, click **Install the package**.

   Any conflicts that were skipped or left unresolved during the configuration are now resolved automatically.

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



The **Installation result** page contains a report that describes any potential problems that arose as a result of the upgrade process and which require review. The upgrade process performs actions to resolve conflicts automatically and complete the upgrade. However, you must review some of these actions to ensure that there was no unexpected data loss. Usually, the problems are unimportant and the upgrade is successful.



If any errors occurred during the upgrade, you must analyze and resolve them. After you have addressed these errors, you must restart the upgrade either with the default options or, depending on the nature of the problems that you addressed, with the items-only or files-only option selected.

Feedback

When the installation is complete, you can give feedback to the Sitecore development team about your upgrade experience. To do this, on the **Installation result** page, click **Tell us what you think about the upgrade**. The feedback is anonymous and the feedback form does not collect any instance or license-specific information.

Chapter 7

Configure Sitecore

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

This chapter contains the following sections:

- Review Changes and update Configuration Files
- Specify the Server role and the search provider
- Remove deprecated indexes
- Upgrading Solr

7.1 Review Changes and update Configuration Files

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

7.1.1 Helix folder structure

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

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

The folders and files are loaded in the following order:

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

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

7.1.2 Configuration file changes

In Sitecore XP 9.3.0, some configuration files were changed.

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

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

7.1.3 Patch files with customizations

After you install the upgrade package, you must enable the patch files with the customizations that were generated by the **Update Installation Wizard**.

The patch files are created and stored in the same folder as the original configuration files. They are saved in the following file format: `<filename>.patch.config.disabled`.

To facilitate traceability and debugging, the wizard also saves the original file with your customizations in the following file format: `<filename>.custom.config.disabled`.

For example, if you customized the `Sitecore.Mvc.config` file, during the analysis, the **Update Installation Wizard** creates:

- `Sitecore.Mvc.patch.config.disabled`

  A patch file with your customizations.

- `Sitecore.Mvc.custom.config.disabled`.

  A backup of the customized configuration file.

  Both files are disabled.

To enable the patch files with the customizations that were generated by the **Update Installation Wizard**, remove the `.disabled` extension from the file name.

Note
It is best practice to store all the generated patch files in the `\Website\App_Config\Include\Z.Custom` folder.

7.1.4 Review the Web.config customizations

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

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

To enable your customizations:

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

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

7.2 Specify the Server role and the search provider

Sitecore XP 9.0.0 introduced the new rule-based configuration functionality that you use to specify the server role performed by each Sitecore instance in your solution. With this functionality, you no longer need to disable and enable configuration files to configure a Sitecore role.

All the Sitecore configuration files are enabled by default. The server role and the search provider are controlled by settings in the `Web.config` file.

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

To specify the server role:

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

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

  The following server roles are supported:

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

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

7.2.1 The Indexing sub-role

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

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

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

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

To specify the search provider:

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

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

The following values are supported:

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

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

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

Upgrading the Solr search provider is described later in this document.

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

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

7.3 Remove deprecated indexes

Important
If you are upgrading from Sitecore XP 8.1 – Sitecore 8.2 Update 7 to Sitecore XP 9.3.0, you must perform this step.

If you are upgrading from Sitecore XP 9.0.0 – 9.0.1 to Sitecore XP 9.3.0, skip this step.

You must remove the following deprecated indexes from your search provider:

- `social_messages_master`
- `social_messages_web`
- `sitecore_list_index`
- `sitecore_analytics_index`

To remove the indexes in Solr:

1. Navigate to the path where your Solr server is installed, for example: `C:\solr-8.1.1\server\solr`
2. Delete any folder that is named after any of the deprecated search indexes.

To remove the indexes in Microsoft Azure:

1. Log in to the Azure portal.
2. Open the Azure Search service.
3. Navigate to the search index, select the relevant indexes and then click **Delete**.

7.4 Upgrading Solr

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

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

1. Upgrade the Solr environment to the new version.

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

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

   Note
   Do not use the Solr IndexUpgraderTool to migrate your old cores or collections to. You must rebuild them and rewrite the data.

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

3. Configure the upgraded Sitecore XP and xConnect instances to use the new Solr deployment.

4. Rebuild all the indexes.

   This is described in the post-upgrade steps.

Content search

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

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

xConnect search

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

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

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

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

Chapter 8

Post-upgrade steps

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

This chapter contains the following sections:

- General maintenance
- Enable telemetry reporting in production
- xDB maintenance
- Recompile your solution

8.1 General maintenance

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

- Clear the browser cache.
- Republish your website.

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

  If you had been using a client language other than English before the upgrade, you must download and import the latest translations from the *Client Translations* section of the Release page. Client translations are available in:
  - Danish (da-DK)
  - German (de-DE)
  - Japanese (ja-JP)
  - Chinese (zh-CN)

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

8.1.1 Upgrade the Session database

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

1. Stop all the Sitecore XP instances that use the same *Session* database.

   If you are using the MongoDB session state provider, drop the *Session* database.

   If you are using the SQL Server session state provider, replace the *Session* database with a clean one from the Sitecore XP installation package.

   If you are using the Redis cache, clean the cache using Redis Cache service "Console".
2. Start all the Sitecore XP instances.

8.1.2 Cache configuration

This section is *only* relevant if you have upgraded from any version of Sitecore XP 8.1 to Sitecore XP 9.3.0.

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.

8.1.3 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 *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*

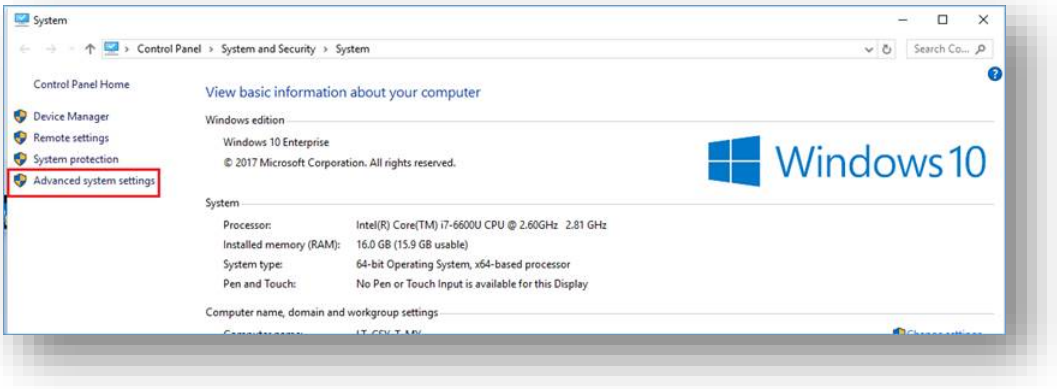
8.2 Enable telemetry reporting in production

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

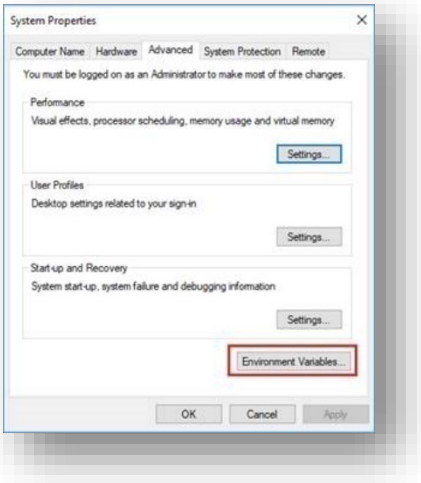
The telemetry system assumes Sitecore is running in a non-production environment by default. When you set up a production environment, you must therefore define a system variable to report the environment type.

To define a system variable:

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

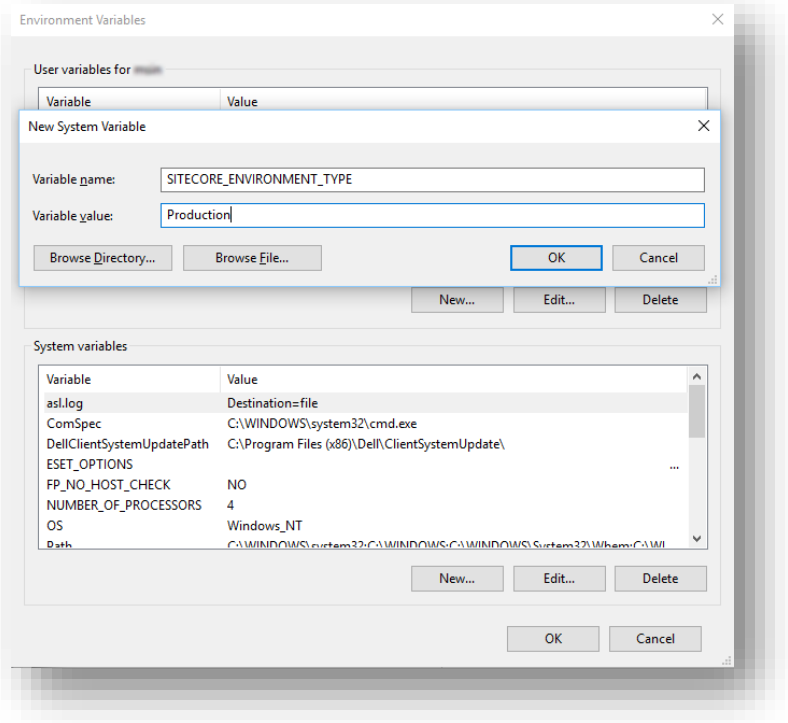


2. In the **System Properties** dialog, click **Environment Variables**.



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

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



5. Restart the computer.

8.3 xDB maintenance

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

8.3.1 Resume the xConnect services

You must start the xConnect services that you stopped in the Upgrade Sitecore xConnect chapter. These services must be started before you enable xDB.

To start the xConnect services:

1. Recycle the IIS xConnect site Application Pool.
2. Start the *Sitecore xConnect Search Indexer* Windows service.
3. Start the *Sitecore Marketing Automation Engine* Windows service.
4. Start the *Sitecore Cortex Processing Engine* Windows service.
5. Rebuild the xConnect indexes.
6. Ensure that all xConnect services are running as expected by reviewing exceptions in the log files.

8.3.2 Enable xDB and Experience Analytics

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

You must now enable the Sitecore Experience Database (xDB) and the Experience Analytics functionality that you disabled when you upgraded the databases.

To enable xDB and Experience Analytics:

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

8.3.3 Using Email Experience Manager (EXM)

Important
You can only use EXM when xDB is enabled.

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

EXM is enabled by default.

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

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

The supported values are:

- yes – default value
- no

Specify the Email Delivery Provider

To use EXM, you must specify the email delivery provider that is used by your solution.

For more information about configuring email delivery, see Sitecore Email Cloud compared to the custom SMTP.

Enable the Transport Layer Security Protocol

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

For more information about enabling TLS 1.2, see Microsoft's documentation.

8.3.4 Redeploy the Marketing Definitions

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

To redeploy the marketing definitions:

1. On the Sitecore Launchpad, click **Control Panel**, **Analytics**, and then click **Deploy Marketing Definitions**.
2. In the **Deploy marketing definitions** dialog box, select all the definitions and taxonomies and click **Deploy**.

8.3.5 Update Sitecore.Analytics.ExcludeRobots.config file

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

Sitecore uses the device detection component to identify robots and the user 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.

8.3.6 Customization of MarketingDefinitionAliasValidationRegex application

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>
```

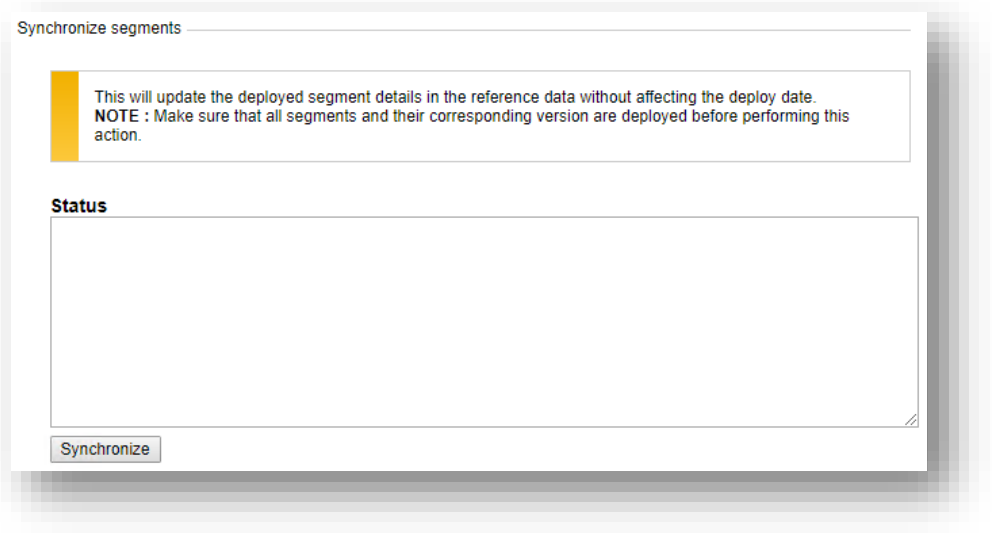
For more information about this step, see the section *Validate the names of the marketing definition items*.

8.3.7 Synchronizing Experience Analytics with Reference Data

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

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

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



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

8.3.8 Upgrading from Sitecore XP 8.1 – 8.2.7

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 these same marketing definitions and marketing taxonomies again.

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

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

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

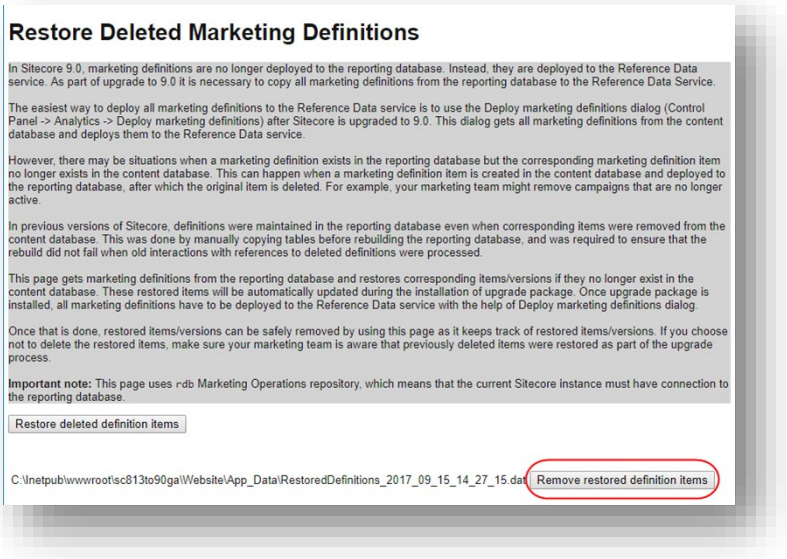
- In a text editor, in the `\App_Data` folder, open the `RestoredTaxonomies_{timestamp}.dat` file.

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

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

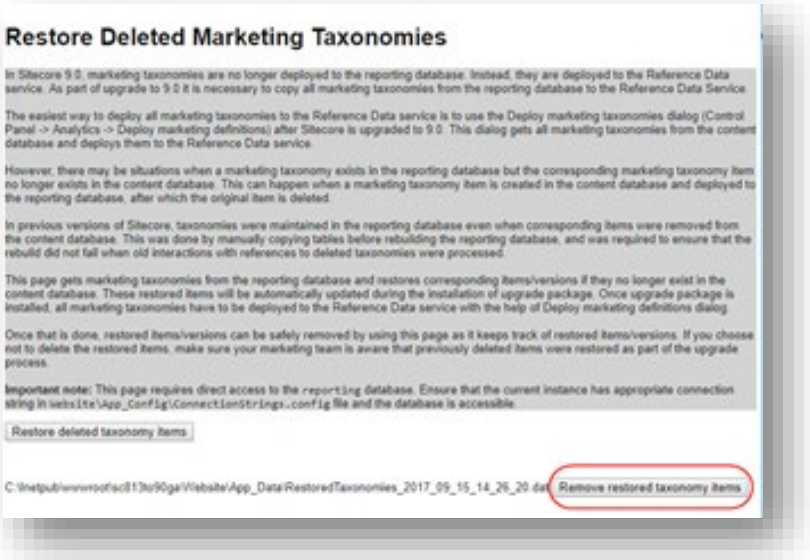


To delete unwanted marketing taxonomy items from the *Master* database:

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, 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 are shipped with Sitecore have been updated already.

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

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

8.3.9 Upgrading from Sitecore XP 9.0.0 and 9.0.1

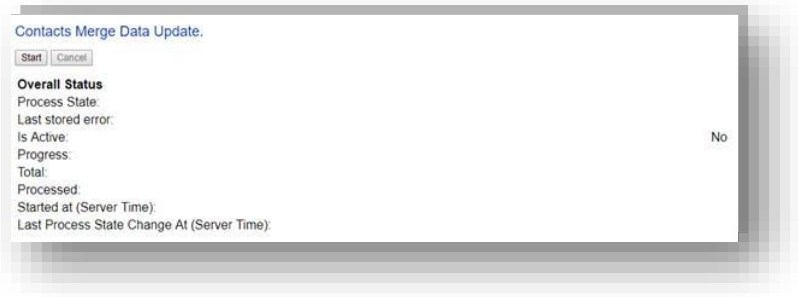
Upgrade the Contact Merge Data

Important
If you are upgrading from Sitecore XP 9.0.1, you must perform this step.

In Sitecore XP 9.0 Update 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 run for a long time and you can run it in a production environment while the system is live. You can also postpone this step, but it is an important part of the upgrade process and you *must* perform it.

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

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.

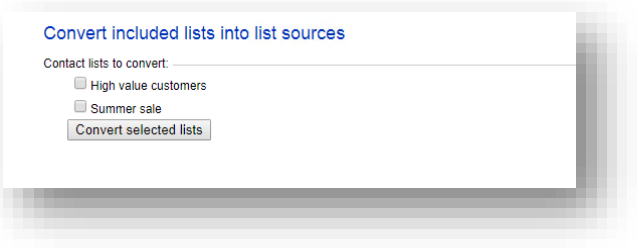
The administration page at `\Website\sitecore\admin\xdb\ContactsMerge.aspx` reports the errors noted in the log file.

Upgrade the included lists in List Manager

In Sitecore XP 9.0.1 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 Content Management instance, open the `/sitecore/admin/ConvertContactListSources.aspx` page.



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

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

Upgrade the Experience Analytics data

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

To upgrade Experience Analytics data:

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

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

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

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:

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

- `[dbo].Fact_GeoMetrics`

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

8.3.10 Rebuild the Reporting database.

After you have upgraded from Sitecore XP 8.1.0 – 9.0.1, 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.

8.4 Recompile your solution

You must recompile your solution with the updated Sitecore DLLs and fix any breaking changes.

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

Chapter 9

Sitecore Identity Server

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

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

This chapter contains the following sections:

- Install Sitecore Identity Server

9.1 Install Sitecore Identity Server

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

Before you install Sitecore Identity Server, you must install .NET Core 2.1.3 Runtime or later.

To install Sitecore Identity Server:

1. On the Sitecore Identity download site, download the following resources:
   - Sitecore Identity - `Sitecore.IdentityServer.4.0.0-r00257.scwdp.zip`
   - Deployment Configuration files - `IdentityServer Deployment Configuration 4.0.0.zip`
2. Create a folder named `c:\resourcefiles` and copy both ZIP files to the folder.
3. Unpack the `IdentityServer Deployment Configuration 4.0.0.zip` file and copy the extracted files to the `c:\resourcefiles` folder.

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

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

#update SIF module
Update-Module SitecoreInstallFramework

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

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

Install-SitecoreConfiguration @identityParams -Verbose
```

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

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

```
.\install.ps1
```

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

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

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

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

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

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

9.1.1 Configure Sitecore Identity Client

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

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

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

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

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

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

Chapter 10

Upgrade Multiple Instances

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

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

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

This chapter contains the following section:

- Upgrade a scaled environment with multiple instances

10.1 Upgrade a scaled environment with multiple instances

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

Content Management Servers

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

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

Content Management, Delivery, Processing, and Reporting Servers

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

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

Scaled XConnect Environment

If you are running Sitecore XP 9.0.0 - Sitecore XP 9.0.1 with the Sitecore XP1 scaled topology, you must perform all the upgrade steps described in the chapter *Upgrade the Sitecore xConnect* Server in alignment with the instructions in the Sitecore scaling scenarios documentation.

Chapter 11

Migrate xDB data

Important
If you are upgrading from Sitecore XP 8.1.0 – 8.2.7 to Sitecore XP 9.3.0, you must perform this step.

If you are upgrading from Sitecore XP 9.0.0 – 9.0.1 to Sitecore XP 9.3.0, 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), but you do not have to migrate data, then you do not have to use the xDB Data Migration Tool.

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

This chapter contains the following section:

- Use the xDB Data Migration Tool to Migrate xDB data

11.1 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 9.3.0 rev. 003498 (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:

   - `sitecore\App_Config\ConnectionStrings.config`
   - `xconnect\App_Config\ConnectionStrings.config`
   - `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.

11.1.1 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 12

Appendix A: Scripts

xConnect application preparation script

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

To use the script:

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

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

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

del App_Data\jobs\continuous\AutomationEngine /q

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

del App_Data\jobs\continuous\IndexWorker /q

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

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

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

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

User credential script for database deployment

This script illustrates how to specify user credentials when you deploy a database.

To create the logins, update the highlighted parameters in the following script and run it for the databases:

```
: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
```

Chapter 13

Appendix B: Generating shard keys and missed percentile values

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

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

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

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

13.1 Run the Sitecore Xdb Collection Update tool

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

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

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

Download the *Sitecore Xdb Collection Update* tool from the Sitecore Experience Platform download site.

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

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

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

13.1.1 Run the tool

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

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

Note
We recommended that you run this tool on offline databases.

If it is impossible to take the database offline, we recommend that you turn off change tracking for the tables whose shards will be updated – *Contacts*, *ContactFacets*, *Interactions*, *InteractionFacets*.

Important
Do not turn off change tracking at database level.

13.1.2 Update operations

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

The update operations are:

- *deviceProfiles* – Updates the *shard key* column in the *device profile* table. The value is calculated based on the device profile ID.
- *deviceProfileFacets* – Updates the *shard key* column in the *device profile facets* table. The value is calculated based on the device profile ID.

- *contacts* – Updates the *shard key* column in the *contact*, *contact's identifiers*, *contact's facets*, *contact's interactions*, and *interaction's facets* tables. The shard key value is calculated based on the contact ID for all these tables and should be the same for all contact-related data.
- *contactIdentifierIndexes* – Updates the *shard key* column for the *contact identifier index* table. The value is calculated based on the contact ID.
- *contactPercentile* - Updates the *percentile* column of the *contact* table. The value is calculated based on the contact ID.
- *interactionPercentile* - Updates the *percentile* column of the *interaction* table. The value is calculated based on the interaction ID.

The update operations are listed in the recommended run order.

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

13.1.3 Tool arguments

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

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

Example:

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