

Sitecore Experience Platform Upgrade Guide

Sitecore Experience Platform 9.0

Upgrade to Sitecore 9.0 from Sitecore 8.1 or later



sitecore[®]
Own the experience[™]

Sitecore Experience Platform Upgrade Guide

Table of Contents

Chapter 1	Introduction	3
Chapter 2	Prerequisites	4
2.1	Prerequisites for Upgrading From Sitecore XP 8.1 or Later	5
Chapter 3	Prepare for the Upgrade	6
3.1	Preparing for the Upgrade	7
3.1.1	Content Testing	7
3.1.2	Restore Deleted Marketing Taxonomies and Marketing Definitions	7
3.1.3	Validate the Names of the Marketing Definition Items	8
3.1.4	Disable the xDB and Sitecore Modules	8
3.1.5	Upgrade the databases	9
3.1.6	Deploy the new databases	9
Chapter 4	Install the Upgrade Package	10
4.1	Use the Update Installation Wizard to Install the Upgrade Package	11
4.2	The Upgrade Package Analysis	13
4.2.1	Resolve Breaking Changes in Custom Code	13
4.2.2	Resolve Configuration File Conflicts	14
Chapter 5	Finalize the Upgrade	18
5.1	Review Custom Changes in Configuration Files	19
5.2	Specify the Server Role	20
5.3	Define the Search Provider	21
5.4	Remove Deprecated Indexes	22
5.5	Update Solr	23
Chapter 6	Upgrade the Modules	25
6.1	Upgrade the Web Forms for Marketers Module	26
Chapter 7	Post-upgrade Steps	27
7.1	General Maintenance	28
7.2	Install and Configure Sitecore xConnect	29
7.3	Enable xDB	30
7.4	Redeploy the Marketing Definitions	31
7.5	Delete unwanted marketing definitions and marketing taxonomies	32
7.6	Update customizations	34
7.6.1	Upgrade Experience Analytics	34
7.6.2	Custom segments	34
7.6.3	Custom reports	34
7.7	Cache Configuration	35
Chapter 8	Upgrade Multiple Instances	36
8.1	Upgrade a Scaled Environment With Multiple Instances	37
Chapter 9	Migrate xDB data	38
9.1	Use the xDB Data Migration Tool to Migrate xDB data	39

Chapter 1

Introduction

This guide contains a detailed description of how to upgrade from Sitecore Experience Platform 8.1 or later to Sitecore Experience Platform 9.0 rev. 171002.

If you are upgrading from Sitecore XP 8.1 rev. 151003 (Initial Release) or later, you can upgrade directly to Sitecore XP 9.0 rev. 171002.

However, if your solution is based on an earlier version of Sitecore, you must first upgrade to Sitecore XP 8.1 rev. 151003 (Initial Release) before you can upgrade to Sitecore XP 9.0.

This guide contains the following chapters:

- Introduction
- Prerequisites
- Prepare for the Upgrade
- Install the Upgrade Package
- Finalize the Upgrade
- Upgrade the Modules
- Post-upgrade Steps
- Upgrade Multiple Instances
- Migrate xDB data

Chapter 2

Prerequisites

This chapter describes the prerequisites required to upgrade to Sitecore XP 9.0 rev. 171002.

This chapter contains the following section:

- Prerequisites for Upgrading From Sitecore XP 8.1 or Later

2.1 Prerequisites for Upgrading From Sitecore XP 8.1 or Later

The following are the prerequisites for upgrading from Sitecore XP 8.1 or later:

- [Sitecore 8.1 rev. 151003 \(Initial Release\) or later](#)
- Sitecore 9.0.0 rev. 171002.update
- Sitecore Update Installation Wizard 3.0.0 rev. 170919.zip
- Sitecore 9.0.0 rev. 171002 (config files).zip

This file contains:

- o Database Upgrade Script.zip

This file contains the following database upgrade scripts:

- o SXP90_BeforeInstall.sql
- o CMS9.0_BeforeInstall.sql
- o CMS90_db_core_BeforeInstall.sql

- o Disable xDB.zip

This file contains the following:

- o Disable.xDB.config

- o MarketingDefinitionsUpgrade.zip

This file contains:

- o RestoreDeletedMarketingTaxonomies.aspx
- o RestoreDeletedMarketingDefinitions.aspx
- o DefinitionItemsNameValidator.aspx

- Sitecore 9.0.0 rev. 171002.zip

In the \Databases folder, this file contains:

- o Sitecore.Experienceforms.dacpac
- o Sitecore.Processing.tasks.dacpac

- [xDB Data Migration Tool](#) – required if you have existing analytics data in MongoDB that you want to use with Sitecore XP 9.0.

Note

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

The upgrade scripts contain changes that are applied to Sitecore databases across every version of Sitecore to ensure compatibility. They also modify certain tables to support new functionality or to increase performance. The changes are applied once.

Chapter 3

Prepare for the Upgrade

To upgrade to Sitecore XP 9.0 rev. 171002, you must do some tasks as preparation. For more information about breaking changes and new features, see the release notes.

This chapter contains the following sections:

- Preparing for the Upgrade

3.1 Preparing for the Upgrade

To prepare for the upgrade, you must perform the following procedures. As part of the upgrade process, some solutions can require redesigning.

3.1.1 Content Testing

Before you upgrade your Sitecore installation, you must stop any content tests that are currently running. This prevents the start date of the tests from being overwritten with the upgrade date.

3.1.2 Restore Deleted Marketing Taxonomies and Marketing Definitions

If you are using the xDB functionality, (when 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 definitions and marketing taxonomies that you have deleted from your Sitecore installation.

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

To restore deleted marketing taxonomy items:

1. Unpack the `MarketingDefinitionsUpgrade.zip` file.
2. Copy the `RestoreDeletedMarketingTaxonomies.aspx` page to the `\sitecore\admin` folder of your website.
3. To open the **Restore Deleted Marketing Taxonomies** page, enter the following URL in your web 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.

4. Click **Restore deleted taxonomy items**.

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

To restore deleted marketing definition items:

1. Unpack the `MarketingDefinitionsUpgrade.zip`.
2. Copy the `RestoreDeletedMarketingDefinitions.aspx` file to the `sitecore/admin` folder of your website.
3. To open the **Restore Deleted Marketing Definitions** page, enter the following URL in your web 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.

4. Click **Restore deleted definition items**.

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

3.1.3 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. To open the **Definition Items Name Validator** page, enter the following URL in your web browser:
<http://<hostname>/sitecore/admin/DefinitionItemsNameValidator.aspx>
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 any 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.

3.1.4 Disable the xDB and Sitecore Modules

To ensure that the upgrade process is not interrupted, you must disable the Sitecore Experience Database (xDB) functionality and all the modules that are running on the Sitecore instance, including Web Forms for Marketers and Email Experience Manager.

Note

The indexing process takes place during the upgrade. Depending on the version of Sitecore you are upgrading from, the number of items can be changed by the update, and this can result in a longer upgrade time.

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

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

To disable the Sitecore Experience Database:

1. Unpack the `Sitecore 9.0.0 rev. 171002 (config files).zip` file and the `Disable xDB.zip` file.
2. In the `\App_Config\Include` folder, create a new `\Z.Custom` folder and copy the `Disable.xDB.config` file to the `\App_Config\Include\Z.Custom` folder.

Important

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

To disable the Email Experience Manager module:

1. Disable all the configuration files in the following folders by adding `.disabled` to the file extension:
 - o `\App_Config\Include\EmailExperience`
 - o `\App_Config\Include\Z.EmailExperience`

Sitecore Experience Platform Upgrade Guide

- o `\App_Config\Include\ExperienceProfile\Sitecore.ExperienceProfile.Reporting.Sitecore.EmailExperience.ExperienceProfile.config`
- 2. In the `\bin` folder of your website, disable all the files that begin with `Sitecore.EmailCampaign`, `Sitecore.EDS`, or `Sitecore.ExM` by moving them outside the `\bin` folder.

To disable the Web Forms for Marketers module:

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

Important

After the upgrade of Sitecore Experience Platform is completed, you must also upgrade the modules. For more information, see the *Upgrade the Modules* chapter in this guide.

3.1.5 Upgrade the databases

To upgrade the databases on solutions that are running on Sitecore XP 8.1 or later:

1. Execute the `CMS9.0_BeforeInstall.sql` script for the Core, Master, and Web databases.
2. Execute the `CMS90_db_core_BeforeInstall.sql` script for the Core database only.
3. If you are using the xDB functionality, you must execute the `SXP90_BeforeInstall.sql` script for the Reporting database.

3.1.6 Deploy the new databases

Starting with Sitecore 9.0, the format of all the databases has changed and are now distributed as Data-tier Application Component Packages (DACPAC), and you can find them in the `Sitecore 9.0.0 rev. 171002.zip` package, located in the `\Databases` folder.

Deploying the new databases allows you to use the Forms component and the Sitecore Experience Database (xDB) functionality.

To use the 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" />
```

To use the Sitecore Experience Database 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" />
```

Chapter 4

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:

- Use the Update Installation Wizard to install the upgrade package
- The upgrade package analysis

4.1 Use the Update Installation Wizard to Install 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 3.0.0 rev.170919.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 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 Sitecore 9.0 rev. 171002.update file. Select the file, and then click **Open**.
6. 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**.

Select an upgrade path

This is an aggregated upgrade package. We were not able to automatically determine which upgrade package should be installed. Please select the upgrade path that is appropriate for your solution.

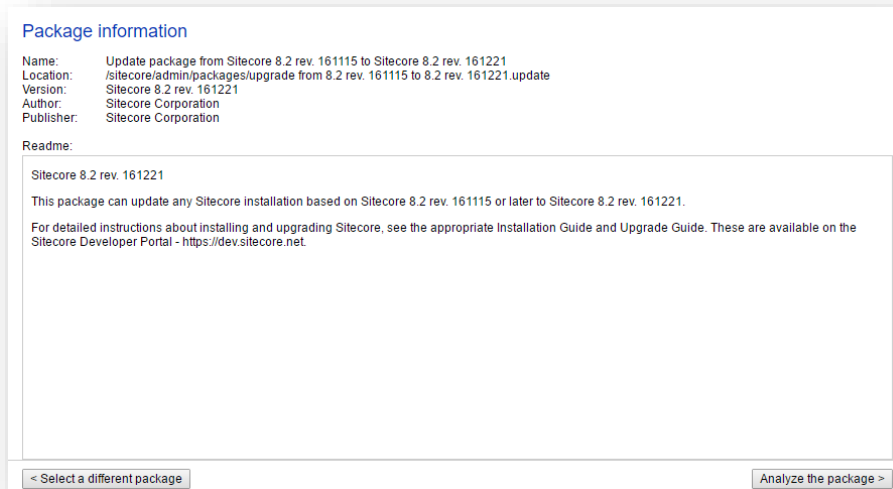
Select the Sitecore version that your solution is based on:

- ☒ Sitecore 8.1 rev. 151003
- ☐ Sitecore 8.1 rev. 151207
- ☐ Sitecore 8.1 rev. 160302
- ☐ Sitecore 8.1 rev. 160519
- ☐ Sitecore 8.1 rev. 160729
- ☐ Sitecore 8.2 rev. 161115

< Select a different package View package information >

Sitecore Experience Platform Upgrade Guide

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



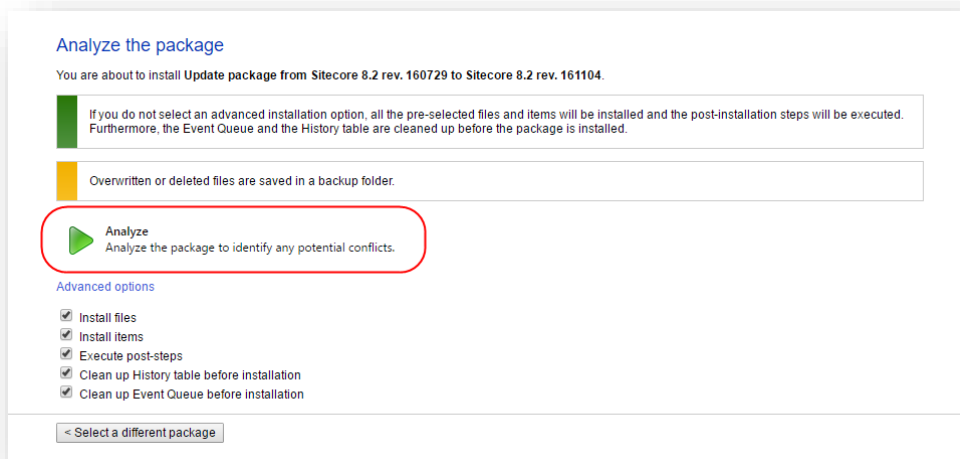
8. On the **Analyze the package** page, click **Advanced options**, review the installation options, and then click **Analyze**.

Note

In the **Advanced options** section, it is recommended that you always select the **Execute post-steps** check box, regardless of the other check boxes you may 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.



9. After the analysis process is complete, click **Analysis result**.
10. If the **Update Installation Wizard** does not find any conflicts, click **Install the package**, and skip to *Finalize the Upgrade* chapter in this guide.

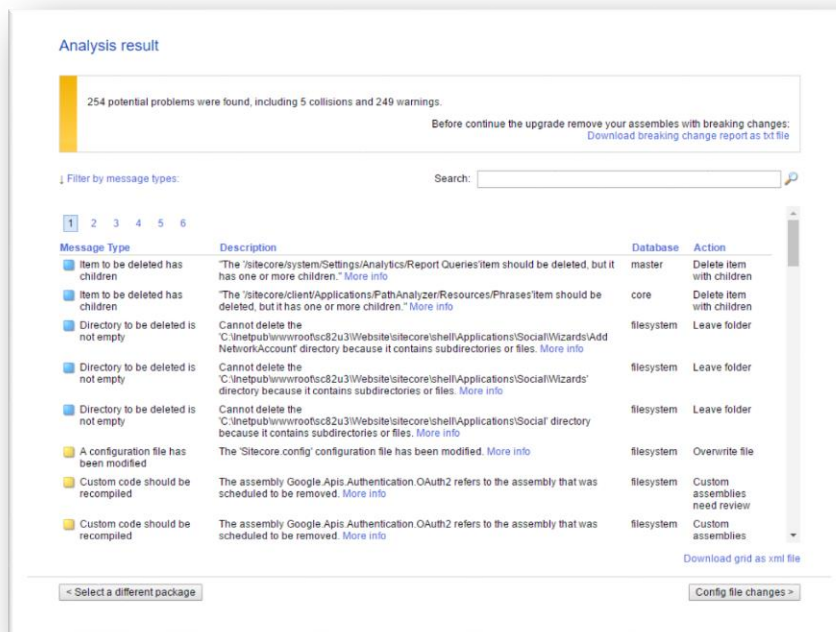
4.2 The Upgrade Package Analysis

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

Collisions 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.



It is recommended that you review all the collisions and warnings, as well as the actions that will be performed to resolve them.

Important

While the majority 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.

4.2.1 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 major Sitecore version, the **Analysis result** page displays a warning.

Sitecore Experience Platform Upgrade Guide

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

Custom code should be recompiled	Type Lucene.Net.Search.Vectorhighlight.VectorHighlightMapper from assembly Lucene.Net.Contrib.FastVectorHighlighter inherited from the type that has got breaking changes. More info	filesystem	Custom assemblies need review
Custom code should be recompiled	Type Lucene.Net.Index.TermPositionVector defined in assembly Lucene.Net has got breaking changes. Custom type Lucene.Net.Search.Vectorhighlight.VectorHighlightMapper defined in assembly Lucene.Net.Contrib.FastVectorHighlighter derives from the changed type and might be affected.	filesystem	Custom assemblies need review
Custom code should be recompiled	Type Lucene.Net.Index.Memory.MemoryIndex+MemoryIndexReader+MemoryTermPositionVector from assembly Lucene.Net.Contrib.Memory inherited from the type that has got breaking changes. More info	filesystem	Custom assemblies need review
Custom code should be recompiled	Type Lucene.Net.Index.Memory.MemoryIndex+MemoryIndexReader+MemoryTermPositions from assembly Lucene.Net.Contrib.Memory inherited from the type that has got breaking changes. More info	filesystem	Custom assemblies need review

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.
- Or
- If you are sure that the affected code is not used during Sitecore 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 all the custom code was removed.

4.2.2 Resolve Configuration File Conflicts

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

Analysis result			
2 potential problems were found, including 2 warnings.			
Search: <input type="text"/>			
Message Type	Description	Database	Action
A configuration file has been modified	The 'Sitecore.config' configuration file has been modified.	filesystem	Overwrite file
A configuration file has been modified	The location of the data folder has been manually modified in the 'Sitecore.config' configuration file. "File path: 'C:\inetpub\wwwroot\NewInstance1\Website\App_Config\Sitecore.config' The 'dataFolder' variable has been manually modified in Sitecore.config configuration file. As a result of this change, the configuration file cannot be upgraded automatically when the update package is installed. The custom path to the data folder will be extracted to a separate include patch file. The custom include patch file is saved as: C:\inetpub\wwwroot\NewInstance1\Website\App_Config\Include\DataFolder.config"	filesystem	Extract customization
Download grid as xml file			
< Select a different package		Config file changes >	

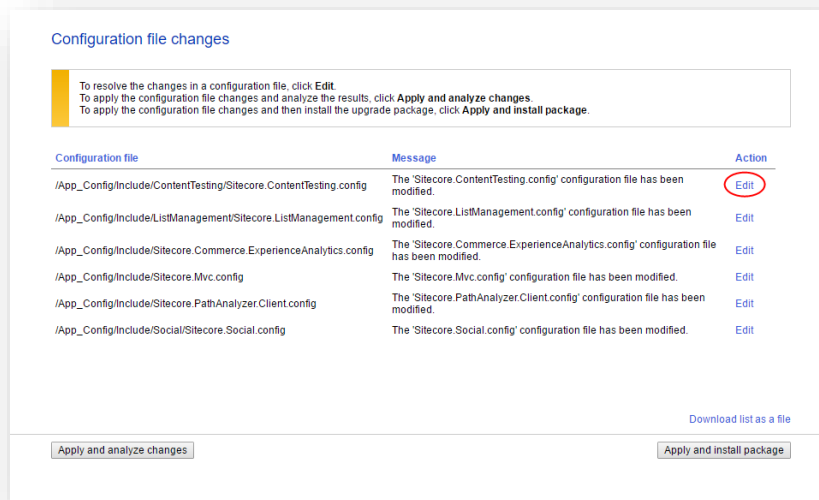
Some of these configuration file conflicts can be resolved automatically and others must be resolved manually.

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

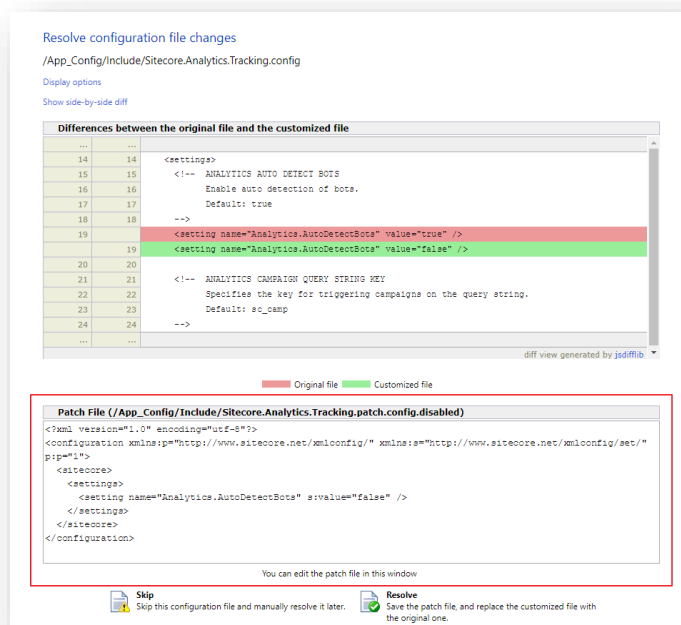
Sitecore Experience Platform Upgrade Guide

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.

Sitecore Experience Platform Upgrade Guide

Note

When you skip the file, the **Update Installation Wizard** automatically detects any customizations in the Sitecore config files, creates the patch, and replaces the customized file with the new file.

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 the old patches 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.

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

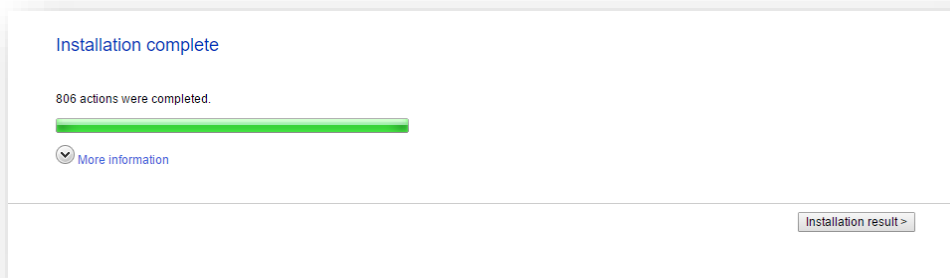
Note

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

9. If the **Update Installation Wizard** does not find any more configuration file conflicts, click **Install the package**.

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

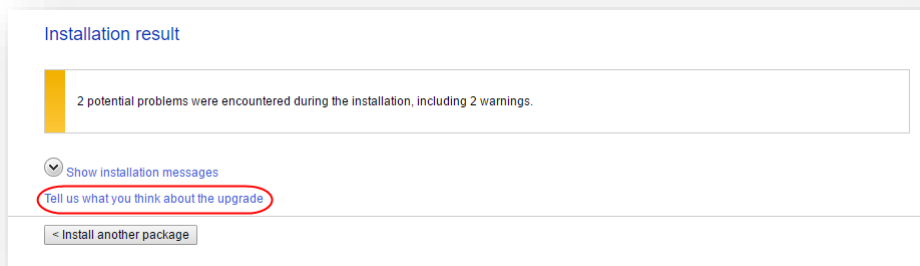
10. When the update process is finished, on the **Installation complete** page, click **Installation result**.



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

Sitecore Experience Platform Upgrade Guide

there was no unexpected data loss. Usually, the problems are inconsequential 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 5

Finalize the Upgrade

After the upgrade package installation is complete, you must finalize the upgrade by reviewing custom changes in configuration files and performing the procedures in this chapter.

This chapter contains the following sections:

- Review Custom Changes in Configuration Files
- Specify the Server Role
- Define the Search Provider
- Remove Deprecated Indexes
- Update SolrResolve Configuration File Conflicts

5.1 Review Custom Changes in Configuration Files

When the upgrade is complete, you must enable the patch files with the customizations that were generated by the **Update Installation Wizard**.

For example, during analysis, if the `Sitecore.Mvc.config` file has been customized, the **Update Installation Wizard** creates a patch file with your customizations and creates a backup of the config file in a disabled state.

The patch files are created and stored in the same folder where the original configuration files are located, and 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`.

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

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 original file with custom changes is saved as `<filename>.custom.config.disabled`.

To replace the original file with the merged file for files that could not be patched:

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

Note

It is best practice to store all the generated patches in the `\Website\App_Config\Include\Z.Custom` path.

5.2 Specify the Server Role

Sitecore 9.0 introduces the new rule-based configuration functionality that you use to define the server role performed by each Sitecore instance in your solution. With this functionality, you no longer have to disable and enable configuration files to configure a Sitecore role.

All Sitecore configuration files are enabled by default and the server role is controlled by one setting 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, add the relevant server role:

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

The supported values for the server role are:

- ContentDelivery
- ContentManagement
- Processing
- Reporting
- Standalone

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

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

5.3 Define the Search Provider

Following the new rule-based configuration functionality, you must define the search provider used for your solution.

To define the search provider:

- In the `\Website\Web.config` file, set the search provider as follows:

```
<add key="search:define" value="Solr" />
```

The following values are supported:

- Lucene
- Solr
- Azure

5.4 Remove Deprecated Indexes

Starting with Sitecore 9.0, 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 Lucene:

1. Navigate to `C:\inetpub\wwwroot\<SitecoreInstanceName>\Data\indexes`
2. Delete any folder named after any of the deprecated search indexes listed at the beginning of this section.

To remove the indexes in Solr:

1. Navigate to the path where your Solr server is installed, for example: `C:\solr-6.6.1\server\solr`
2. Delete any folder named after any of the deprecated search indexes listed at the beginning of this section.

To remove 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 index(es) and then click **Delete**.

5.5 Update Solr

To continue using the Solr search provider in Sitecore 9.0, you must update your existing Solr instance to version 6.6.1. For more information on how to update Solr, see the Solr documentation.

Note

Sitecore XP 9.0 supports the Managed Schema approach. Therefore, it is no longer necessary to set up the Solr Server to run the Classic Schema.

To update Solr cores to use managed schema:

1. Open the managed schema file, located in the `\conf` folder. Depending on your Solr installation, the location of this folder might be different. For example, the default location is: `C:\solr-6.6.1\server\solr\sitecore_core_index\conf`
2. In the `<uniqueKey>id</uniqueKey>` node, set the `uniqueKey` value to `_uniqueid` as follows: `<uniqueKey>_uniqueid</uniqueKey>`.
3. In the `<field>` section, add the field configuration for the `_uniqueid` value as follows:

```
<field name="_uniqueid" type="string" indexed="true" required="true" stored="true"/>
```

4. Save your changes.
5. In the `\conf\Solrconfig.xml` file, delete the following line:

```
<schemaFactory class="ClassicIndexSchemaFactory" />
```

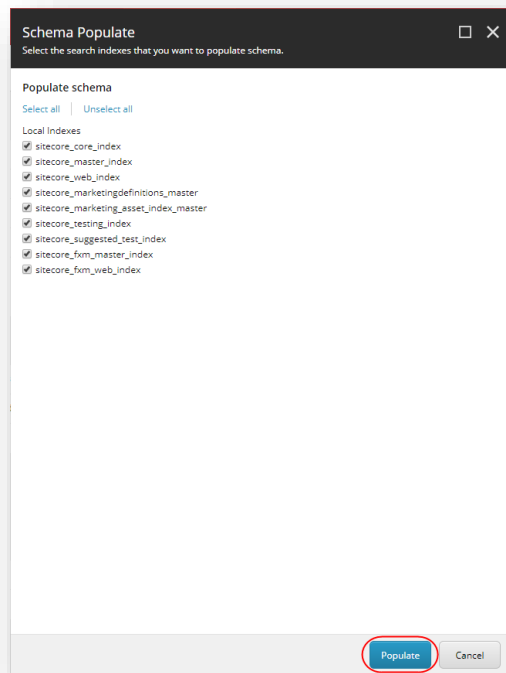
Important

You must repeat steps 1–5 for each core index.

6. During the installation of the update package, all custom Solr configurations was extracted into patch files. You must ensure that all of the patches have been enabled. For more information, refer to the *Review Custom Changes in Configuration Files* section in this guide.
7. On the **Sitecore Launchpad**, click **Control Panel**, and in the **Indexing** section, click **Populate Solr Managed Schema**.

Sitecore Experience Platform Upgrade Guide

8. In the **Schema Populate** dialog box, select all indexes, and then click **Populate**.



Chapter 6

Upgrade the Modules

After you have completed the upgrade of the Sitecore Experience Platform, you must upgrade any Sitecore modules that you have installed.

In this chapter, you find the the upgrade steps for the Web Forms for Marketers module.

For information on how to upgrade other modules, refer to the upgrade documentation for the specific module. You can download all upgrade and installation documentation from the [Sitecore Downloads page](#).

This chapter contains the following section:

- Upgrade the Web Forms for Marketers Module

6.1 Upgrade the Web Forms for Marketers Module

To upgrade the Web Forms for Marketers module, you must download and install the Web Forms for Marketers 9.0 Update Package.

To install the upgrade package:

1. 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>
2. The wizard guides you through the update process and helps you:
 - Upload the upgrade package.
 - Analyze the package.
 - Install the package.
3. After you install the package, you must complete the *Post-upgrade*.

Note

For more information about the Update Installation Wizard, refer to the *Install the Upgrade Package* section in this guide.

Chapter 7

Post-upgrade Steps

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

This chapter contains the following sections:

- **Error! Reference source not found.**
- Install and Configure Sitecore xConnect
- Enable xDB
- Redeploy the Marketing Definitions
- Delete unwanted marketing definitions and marketing taxonomies
- Update customizations
- Cache Configuration

7.1 General Maintenance

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

1. Clear the browser cache.
2. Perform a republish of your site. You must publish the entire site, including system items, templates, and so on, to every publishing target.
3. Rebuild the search indexes:
 - On the **Sitecore Launchpad**, click **Control Panel**, and in the **Indexing** section, click **Indexing manager**. In the **Indexing Manager** dialog box, click **Select all**, and then click **Rebuild**.
4. Rebuild the Link database for the Master and Core databases:
 - On the **Sitecore Launchpad**, click **Control Panel**, and in the **Database** section, click **Rebuild Link Databases**. Select the Master and Core databases and then click **Rebuild**.

7.2 Install and Configure Sitecore xConnect

If you want to use the Sitecore Experience Database (xDB) functionality, you must install and configure Sitecore xConnect before you continue with the post-upgrade.

For more detailed information about installing xConnect — including deployment topologies — see the Sitecore Experience Platform 9.0 Installation Guide. You can download the latest Sitecore Installation Guide from the [Sitecore Downloads Page](#).

To configure Sitecore xConnect:

1. Install xConnect following the instructions in the Sitecore Installation Guide.
2. For a single instance, in the Sitecore instance that you have upgraded, in a text editor, 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;FindValue="/>
<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="/>
```

Note

These connection strings are for a single instance only. For a scaled environment, see [the reference documentation for server role configuration](#) on the [Sitecore Documentation site](#).

Grant permissions to the xConnect certificates

You must grant read permissions to the xConnect client and server certificates to your Sitecore instance(s) `ApplicationPoolIdentity`.

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 you grant *read* permissions to the relevant Sitecore instance Application Pool Identity. For example: `IIS AppPool\<YourSitecoreAppPoolName>`.
5. Restart IIS.

7.3 Enable xDB

You must now enable the Sitecore Experience Database (xDB) functionality that you disabled in the *Disable the xDB and Sitecore Modules* section.

To enable xDB:

- In the `\App_Config\Include\Z.Custom` folder, remove the `Disable.xDB.config` file.

7.4 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**.

Note

If you do not do this, the binary data stored in the **Data** column in the definition tables (*CampaignActivityDefinitions*, *GoalDefinitions*, and so on) might not be compatible with this version of Sitecore.

7.5 Delete unwanted marketing definitions and marketing taxonomies

If you are using the Sitecore Experience Database (xDB) functionality, (for example, the `Xdb.Enabled` setting is set to `true` and xDB is enabled in the license), 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.

Note

If xDB is disabled on your website, you can skip this procedure.

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

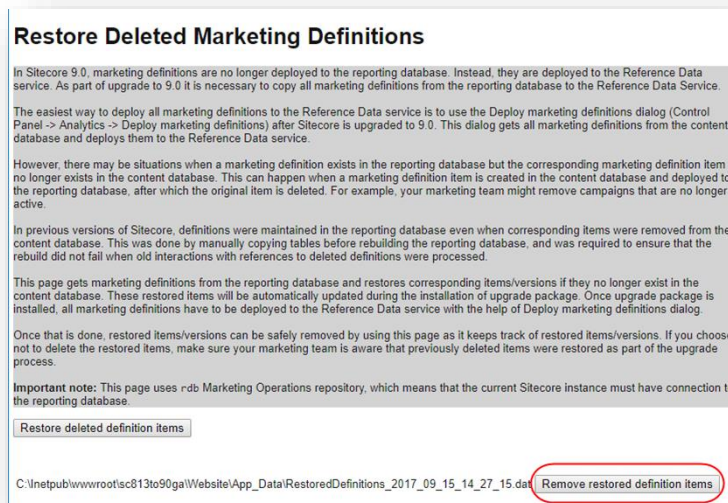
- In a text editor, in the `\App_Data` folder, open the `RestoredDefinitions_{timestamp}.dat` file.

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 your web 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 your web 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**.



7.6 Update customizations

If you customized Experience Analytics or created any custom segments, reports, or dashboards, you must update these.

7.6.1 Upgrade Experience Analytics

In Sitecore 9.0, some subsystems that affect Experience Analytics have been changed. If you have customized Experience Analytics, you must update these customizations to support the updated subsystems.

7.6.2 Custom segments

If you created any custom segments under the default dimensions, you must manually update these 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* on the [Sitecore Documentation site](#).

7.6.3 Custom reports

In Sitecore 9.0 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 report or dashboard.

For more information about adding a parameter to a report, see the walkthrough [Creating an Experience Analytics Report](#) on the Sitecore Documentation site.

7.7 Cache Configuration

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

For more information, see the [cache API changes](#) topic on the Sitecore Documentation Site. This topic is *only* valid if you have upgraded from any version of Sitecore 8.1 to 9.0.

For more information about cache configuration, see the [Cache Configuration Reference](#) document on the Sitecore Developer Network.

Chapter 8

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.

For a scaled environment, you must repeat the steps in this guide for each instance, with the exception of actions related to the database, which you must follow only once for each database .

This chapter contains the following section:

- Upgrade a Scaled Environment With Multiple Instances

8.1 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 follow all the steps described in this guide, with the following differences:

Content Management

Complete all the steps in this guide.

Content Delivery, Processing, and Reporting

Complete all of the steps in this guide, except for the following:

- In the *Prepare for the Upgrade* chapter (Chapter 3), you must only follow the steps in the *Disable the xDB and Sitecore Modules* section.
- In the *Use the Update Installation Wizard to Install the Upgrade Package* section (Chapter 4.1):
 - In the **Analyze the package** dialog box, in the **Advanced options** section, clear the **Install items** check box.
- In the *Specify the Server Role* section (section 5.2):
 - Set the “`role:define`” setting to either the ContentDelivery, Processing, or Reporting instance respectively.
- In the *Upgrade the Modules* chapter (chapter 6):
 - In the **Analyze the package** dialog box, in the **Advanced options** section, clear the **Install items** check box.
- You must also skip all of the steps in the *Post-upgrade* chapter (chapter 7).

Chapter 9

Migrate xDB data

Sitecore XP 9.0 introduces a new data model for the Experience Database (xDB). This means that you must convert all data in the MongoDB database into the new format. You can use the xDB Data Migration Tool to perform this data conversion.

Important

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

9.1 Use the xDB Data Migration Tool to Migrate xDB data

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

You can find the Reporting database in the Sitecore 9.0.0 rev. 171002.zip package, located in the \Databases folder.

To migrate the xDB data:

1. To deploy a new Reporting database, deploy the Sitecore.Reporting.dacpac database, located in the Sitecore 9.0.0 rev. 171002.zip package, in the \Databases folder.
2. In a text editor, in the \App_Config folder of your Sitecore instance, in the ConnectionStrings.config file, specify a new reporting database name to point to. For example:

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

3. After you deploy the new Reporting database to the SQL Server, you can execute the xDB Data Migration tool. To do this, refer to the *xDB Migration Tool Guide* that you can download from the [Sitecore Downloads page](#).