

Sitecore XP Upgrade Guide

Sitecore XP 8.1

Step by step guide to the Sitecore 8.1 upgrade and conversion process



sitecore[®]
Own the experience[™]

Table of Contents

1.1	Prerequisites	3
1.1.1	Sitecore Experience Platform 8.1 requires a new license key	3
1.2	Upgrade Process	4
1.2.1	Content Testing	4
1.2.2	Pre-conditions for installing the Upgrade Package	4
1.2.3	Installing the Upgrade Package	7
1.3	Post-Installation steps for the Upgrade Package	8
1.3.1	Updating Solr Assemblies	8
1.3.2	Updating Configuration Files.....	8
1.3.3	Updating Modules.....	8
1.3.4	Upgrading Multiple Instances.....	12

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2019 Sitecore. All rights reserved.

1.1 Prerequisites

If your solution is based on an earlier version of Sitecore, such as Sitecore CMS 6.x or 7.x, you must first upgrade to Sitecore 8.0 rev.141212 before you can install the Sitecore 8.1 rev.151003 upgrade package. You can download all the prerequisite components from the [Sitecore Developer Portal](#) website.

Prerequisites for running this update:

- Sitecore 8.0 rev.141212 or later
- Sitecore Update Installation Wizard 1.0.0 rev. 150930.zip
- The Sitecore Update Package which can be found on the [Sitecore Developer Portal](#) website
- ASP.NET MVC assemblies
- For Microsoft SQL Server, you also need the following database upgrade scripts:
 - CMS80U3_BeforeInstall.sql
 - SXP81_BeforeInstall.sql
 - CMS81_AfterInstall.sql

To download these files, on the Sitecore Developer Portal, go to the [Sitecore Experience Platform 8.1](#) page, and in the **Upgrade options** section, click *Config files for upgrade*.

Important

When you configure xDB, you must synchronize all the servers in your solution to a single reliable time source, for example by means of the Network Time Protocol (NTP). The aggregation of engagement automation states depends on the system time and changing this can lead to incorrect aggregation results or loss of data.

Note

Upgrading a solution that uses Oracle databases is currently not supported.

1.1.1 Sitecore Experience Platform 8.1 requires a new license key

To enable all the features in the Sitecore Experience Platform 8.1, your license file must contain the new *Sitecore.xDB* key. If your license file does not contain this key, Sitecore will default to Experience Management (CMS-only) mode. Any customers or partners who have a license for the Sitecore Experience Platform but do not have this key should contact their account manager or login to the [Sitecore Partner Network](#).

1.2 Upgrade Process

1.2.1 Content Testing

Before you upgrade your Sitecore installation, we recommend that you stop all the content tests that you are running. Stopping the content tests prevents the start date of the tests from being overwritten with the upgrade date.

1.2.2 Pre-conditions for installing the Upgrade Package

To ensure that the upgrade process is not interrupted:

1. If you have also installed Email Experience Manager (EXM), Web Forms for Marketers (WFFM) or the IP Geolocation Service application, disable the following configuration files by adding `.disabled` to the file extension:
 - o `Sitecore.EmailExperience.AnalyticsProcessing.config`
 - o `Sitecore.EmailExperience.ContentDelivery.config`
 - o `Sitecore.EmailExperience.ContentManagement.config`
 - o `Sitecore.EmailExperience.ContentManagementPrimary.config`
 - o `Sitecore.EmailExperience.Core.config`
 - o `Sitecore.EmailExperience.Reporting.config`
 - o `Sitecore.EmailExperience.UI.config`
 - o `Sitecore.Forms.config`
 - o `Sitecore.CES.config`
 - o `Sitecore.CES.GeoIp.config`
 - o `Sitecore.CES.GeoIp.LegacyLocation.config`
2. Copy and replace the [ASP.NET MVC assemblies](#)
3. You must implement the pre-installation configuration file changes before the installation begins.

- 1) Disable analytics.

In the `Sitecore.Analytics.config` file, set the `Analytics.Enabled` setting to *false*.

- 2) Disable the `Sitecore.Analytics.Processing.Aggregation.Services.config` and `Sitecore.Analytics.Processing.Services.config` files by renaming them.
- 3) In the `web.config` file, in the `<compilation>` section, in the `<assemblies>` node, remove the version for the `System.Web.Mvc`, `System.Web.Http`, `System.Web.Http.WebHost`, and `System.Net.Http.Formatting` assemblies.

After you implement these changes, the section should look like this:

```
<assemblies>
  <add assembly="System.Web.Abstractions, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
  <add assembly="System.Web.Routing, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
  <add assembly="System.Data.Linq, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
  <add assembly="System.Web.Helpers, Version=3.0.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
  <add assembly="System.Web.Mvc,
Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
  <add assembly="System.Web.WebPages, Version=3.0.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
```

```
<add assembly="System.Web.WebPages.Razor, Version=3.0.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
<add assembly="System.Web.Http,
Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
<add assembly="System.Web.Http.WebHost,
Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
<add assembly="System.Net.Http.Formatting,
Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
</assemblies>
```

- 4) In the `web.config` file, in the `<assemblyBinding>` section, change the `oldVersion` value for the `System.Web.Mvc`, `System.Web.Http`, `System.Web.Http.WebHost`, and `System.Net.Http.Formatting` assembly identities, from `1.0.0.0-5.0.0.0` to `1.0.0.0-5.2.3.0` and the `newVersion` value from `5.1.0.0` to `5.2.3.0`.

After you implement these changes, the section should look like this:

```
<assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
  <dependentAssembly>
    <assemblyIdentity name="Lucene.Net" publicKeyToken="85089178b9ac3181"/>
    <bindingRedirect oldVersion="0.0.0.0-2.9.4.0" newVersion="3.0.3.0"/>
  </dependentAssembly>
  <dependentAssembly>
    <assemblyIdentity name="Newtonsoft.Json"
publicKeyToken="30ad4fe6b2a6aeed"/>
    <bindingRedirect oldVersion="0.0.0.0-6.0.0.0" newVersion="6.0.0.0"/>
  </dependentAssembly>
  <dependentAssembly>
    <assemblyIdentity name="System.Web.Mvc"
publicKeyToken="31bf3856ad364e35" xmlns="urn:schemas-microsoft-com:asm.v1"/>
    <bindingRedirect oldVersion="1.0.0.0-5.2.3.0" newVersion="5.2.3.0"
xmlns="urn:schemas-microsoft-com:asm.v1"/>
  </dependentAssembly>
  <dependentAssembly>
    <assemblyIdentity name="System.Web.WebPages.Razor"
publicKeyToken="31bf3856ad364e35" xmlns="urn:schemas-microsoft-com:asm.v1"/>
    <bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="3.0.0.0"
xmlns="urn:schemas-microsoft-com:asm.v1"/>
  </dependentAssembly>
  <dependentAssembly>
    <assemblyIdentity name="System.Web.Http"
publicKeyToken="31bf3856ad364e35" xmlns="urn:schemas-microsoft-com:asm.v1" />
    <bindingRedirect oldVersion="1.0.0.0-5.2.3.0" newVersion="5.2.3.0"
xmlns="urn:schemas-microsoft-com:asm.v1" />
  </dependentAssembly>
  <dependentAssembly>
    <assemblyIdentity name="System.Web.Http.WebHost"
publicKeyToken="31bf3856ad364e35" xmlns="urn:schemas-microsoft-com:asm.v1" />
    <bindingRedirect oldVersion="1.0.0.0-5.2.3.0" newVersion="5.2.3.0"
xmlns="urn:schemas-microsoft-com:asm.v1" />
  </dependentAssembly>
  <dependentAssembly>
    <assemblyIdentity name="System.Net.Http.Formatting"
publicKeyToken="31bf3856ad364e35" xmlns="urn:schemas-microsoft-com:asm.v1" />
    <bindingRedirect oldVersion="1.0.0.0-5.2.3.0" newVersion="5.2.3.0"
xmlns="urn:schemas-microsoft-com:asm.v1" />
  </dependentAssembly>
  <dependentAssembly>
    <assemblyIdentity name="Ninject" publicKeyToken="c7192dc5380945e7"
xmlns="urn:schemas-microsoft-com:asm.v1"/>
    <codeBase version="3.2.0.0" href="bin\Social\Ninject.dll"
xmlns="urn:schemas-microsoft-com:asm.v1"/>
  </dependentAssembly>
</assemblyBinding>
```

- 5) If you are using the Solr search provider, the `Global.asax` file must be changed. Make a backup of your `Global.asax` file, open the file and replace the `Inherits` value with the following: `"Sitecore.Web.Application"`.

In the `Sitecore.ContentSearch.config` file, comment out the following settings:
`ContentSearch.SearchMaxResults` and
`ContentSearch.DefaultIndexConfigurationPath`.

- 6) In the `Sitecore.ContentTesting.config` file, comment out the following event handler, unless it is already commented out:

```

    <event name="item:saved">
      <handler type="Sitecore.ContentTesting.Events.GenerateScreenshot,
Sitecore.ContentTesting" method="OnItemSaved">
        <excludeFields hint="list:ExcludeFieldFromComparison">
          <created>__created</created>
          <createdby>__created by</createdby>
          <updated> updated</updated>
          <updatedby> updated by</updatedby>
          <revision> revision</revision>
          <sortorder> sortorder</sortorder>
          <validfrom>__valid from</validfrom>
          <workflow>__workflow</workflow>
          <workflowstate> workflow state</workflowstate>
          <lock> lock</lock>
          <pageleveltestsetdefinition> Page Level Test Set
Definition</pageleveltestsetdefinition>
        </excludeFields>
      </handler>
    </event>

```

- 7) In the `Sitecore.Mvc.config` file, after the `<pipelines>` section, insert the following new node:

```

    <mvc>
      <areas>
        <!-- MVC: Sitecore will by default register all areas in the initialize
pipeline.
                Use the below list to specify areas that should not be
registered in the initialize pipeline.
-->
        <excludedAreas>
          <area
type="Sitecore.Social.Client.Mvc.Areas.Social.SocialAreaRegistration,
Sitecore.Social.Client.Mvc" />
        </excludedAreas>
      </areas>
    </mvc>

```

- 8) In the `Sitecore.Speak.config` file, set the "Speak.Components.KnockoutPresenter" setting to the default value "scKoPresenter".

4. Make sure that IIS is configured to allow access to the `/sitecore/admin` folder.

Note

This step may be necessary if you used the `setup.exe` file to install Sitecore. The `setup.exe` file configures IIS to disallow access to the `/sitecore/admin` folder and prevents you from using the Update Installation Wizard.

5. Install the Sitecore Update Installation Wizard 1.0.0 rev.150930

On the Sitecore Desktop click **Sitecore, Development Tools, Installation Wizard** and then install the Sitecore Update Installation Wizard 1.0.0 rev. 150930.zip package.

6. Upgrade the CMS database schemas:

SQL Server: Execute the `CMS80U3_BeforeInstall.sql` script on the *Master*, *Core*, and *Web* databases.

7. Upgrade the SXP database schemas:

SQL Server: Execute the `SXP81_BeforeInstall.sql` script on the reporting databases.

8. In the **Marketing Control Panel**, move any existing *Outcome definitions* to `/sitecore/system/Marketing Control Panel/Outcomes`.

You can group the outcome definitions into folders, if you want to.

In Sitecore 8.1, *Outcome Types* have been replaced by *Outcome groups*.

Note

Ensure that you do not leave any *Outcome definitions* under the *Outcome types*. All the *Outcome types* and their dependent *Outcome definitions* will be deleted at the end of the upgrade process.

1.2.3 Installing the Upgrade Package

Use the Update Installation Wizard to install the upgrade package and update your website from Sitecore 8.0 to Sitecore 8.1.

Before you install the upgrade package:

- Back up your website.

To install the upgrade package:

1. To open the Update Installation Wizard, enter the following URL in your web browser:

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

For more information on how to use the wizard, see the *Update Installation Wizard* guide on the [Sitecore Developer Portal](#) website.

2. Use the Update Installation Wizard to install the Sitecore 8.1 update package.

The wizard guides you through the update process and helps you:

- Upload the update package.
- Analyze the package.
- Install the package.

Note

At the end of the installation process, the Login page will be shown. Please login again and wait for the installation results.

If you are using the *Item Clones* feature and have clones, ensure that you carry out the following steps after you install the upgrade package:

1. Upgrade the CMS database schemas:

SQL Server: Execute the `CMS81_AfterInstall.sql` script on the Master and Core databases.

1.3 Post-Installation steps for the Upgrade Package

After installing the upgrade package, there are a number of steps that must be taken in order to complete the installation of Sitecore 8.1.

1.3.1 Updating Solr Assemblies

If you are using the Solr search provider, you must upgrade the corresponding assemblies. Use the Solr support package and extract the assemblies into the `/bin` folder of the website.

1.3.2 Updating Configuration Files

After the upgrade package has been installed, you must update your configuration files.

1. Apply the configuration file changes described in Sitecore 8.1 rev. 151003 Configuration File Changes document.
2. If you are upgrading from 8.0 rev. 141212 (Initial Release), you must also apply the configuration file changes described in these documents:
 - o Sitecore 8.0 rev. 150121 (Update-1) Configuration File Changes document
 - o Sitecore 8.0 rev. 150223 (Update-2) Configuration File Changes document
 - o Sitecore 8.0 rev. 150427 (Update-3) Configuration File Changes document
 - o Sitecore 8.0 rev. 150621 (Update-4) Configuration File Changes document
 - o Sitecore 8.0 rev. 150812 (Update-5) Configuration File Changes document

All of these documents can be downloaded from the [Sitecore Developer Portal](#).

1.3.3 Updating Modules

Upgrade any modules you have installed to make them compatible with the current latest version of Sitecore. If you are using the EXM or WFFM module, you may also need to enable the following configuration files:

- o `Sitecore.EmailExperience.AnalyticsProcessing.config`
- o `Sitecore.EmailExperience.ContentDelivery.config`
- o `Sitecore.EmailExperience.ContentManagement.config`
- o `Sitecore.EmailExperience.ContentManagementPrimary.config`
- o `Sitecore.EmailExperience.Core.config`
- o `Sitecore.EmailExperience.Reporting.config`
- o `Sitecore.EmailExperience.UI.config`
- o `Sitecore.Forms.config`

Clearing browser cache

After you change the configuration files, you must clear your browser's cache.

Note

These steps may vary depending on the browser and version of the browser that you are using.

Internet Explorer:

1. In Internet Explorer, click **Tools, Internet Options**.
2. In the **Internet Options** dialog box, in the **Browsing History** section, click **Delete**, and then select the **Temporary Internet files and website files** checkbox.

3. Click **Delete**.

Rebuilding Search Indexes and the Link Database

To complete the upgrade process, you must rebuild the search indexes.

To rebuild all the `ContentSearch` indexes:

1. In the Sitecore **Control Panel**, in the **Indexing** section, click **Indexing manager**.
2. In the **Indexing Manager**, in the **Rebuild search index** section, click **Select all** and then click **Rebuild**.

To rebuild the search indexes for the `Quick search index` – also for the *Master* and *Core* databases, if you have them:

1. In the Sitecore **Control Panel**, in the **Database** section, click **Rebuild search index**.
2. In the **Rebuild Search Indexes** dialog box, select the **Quick search index** check box and then click **Rebuild**.

To rebuild the link database for the *Master* and *Core* databases:

1. In the Sitecore **Control Panel**, in the **Database** section, click **Rebuild link databases**.
2. In the **Rebuild Link Databases** dialog box, select the *Master* and *Core* databases and then click **Rebuild**.

Updating Social Data

To upgrade your solution from the Sitecore, 8.0 rev. 141212 (Initial version), you must also update the social profiles data to ensure that it is compliant with Sitecore 8.1.

To update the social profiles data:

1. In your browser, open the following page:
<http://<hostname>/sitecore/admin/UpdateSocialProfiles.aspx>

Note

The tool should be executed on a Content Management server. If you are running a multi-instance environment, you only need to execute it on one of the servers

2. Log in as an administrator.
3. Click **Run**.
4. See the Sitecore logs for details.

Note

If execution fails, you can safely execute the tool several times

Upgrading Content Testing Data

Note

If you are running on xDB Cloud, you do not need to upgrade the content testing data. The `Sitecore.Xdb.config` file is disabled in xDB cloud.

1. In your browser, open the following page:
<http://<hostname>/sitecore/admin/ContentTestingUpgrade.aspx>

Note

Make sure that analytics is enabled.

To enable analytics, in the `Sitecore.Xdb.config` file, set the `Xdb.Enabled` setting to `true`.

2. Log in as an administrator.
3. Click the Start button.

Creating Outcome groups

1. In the **Marketing Control Panel**, select **Outcome groups** and create outcome groups that correspond to the outcome types that you used previously.
2. To map an outcome definition to an outcome group, select the outcome definition item and in the **Classification** section, in the **Group** field, select the appropriate outcome group.

Re-deploy marketing definitions

1. To re-deploy *Outcomes* and *Taxonomies* to the reporting database, in the Launchpad, click **Control Panel, Analytics** and then click **Deploy Marketing Definitions**.
2. Select the definitions and taxonomies that you would like to deploy 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) may not be compatible with Sitecore 8.1.

Redeploy Path Analyzer's maps and trees

1. Make sure that you have a backup of your current reporting database.
2. Launch Path Analyzer's admin page: `/sitecore/admin/pathanalyzer.aspx`
3. In the **Upgrade** section click the **Upgrade** button.
4. You must rebuild the Path Analyzer's data and there are two ways to do this:
 - A: in the **Historic Map Rebuild** section, click **Rebuild**.
This option will only rebuild the Path Analyzer's data and will not have an effect on other parts of the platform.
 - B: Use the `/sitecore/admin/rebuildreportingdb.aspx` page to perform a full rebuild of the *reporting* database.
This step will rebuild the data for the whole platform, including Experience Analytics, Content Testing, and so on.

Note

Option A will spare you from carrying out another data re-stream from xDB.

Option B is only recommended if you intend to rebuild the *reporting* database after upgrading.

Update GeolP entries in xDB

To convert the existing GeolP entries in xDB, run the `HashStoredIPs.exe` tool.

You can download this tool from the [Sitecore Experience Platform 8.1 Initial Release](#) page on the Sitecore Developer Portal.

Rebuilding the reporting database (Optional)

Due to the many changes that have been implemented in the *reporting* database schema, it is not possible to fill all the new columns and tables with accurate data using only the `SXP81_BeforeInstall.sql` script. Source data is often in the *collection* or *Master* database.

Sitecore XP Upgrade Guide

Please be aware of the following peculiarities of the upgrade script:

- The `IsActive` column is added to the `CampaignActivityDefinitions` table.
The default value of this column is set to `TRUE` for all the existing records. While `TRUE` should be the correct value for most of the records, it is better to run the Marketing Definitions Deployment tool to fill this table with accurate data after the update.
You can access the Marketing Definitions Deployment tool from the Sitecore Control Panel in the Analytics section.
- The `IsActive` column is added to the `GoalDefinitions` table.
The default value of this column is set to `TRUE` for all the existing records. While `TRUE` should be the correct value for most of the records, it is better to run the Marketing Definitions Deployment tool.
- The `IsActive` column is added to the `MarketingAssetDefinitions` table.
`TRUE` is assigned as a default value of this column for all existing records. While `TRUE` should be a correct value for most of the records, it is better to run the Marketing Definitions Deployment tool.
- The structure of the `OutcomeDefinitions` table has been changed in Sitecore 8.1.
The upgrade script drops this table and then re-creates it. Run the Marketing Definitions Deployment tool to fill this table with accurate data after the update.
- The `OutcomeDefintionTypes` table has been removed in Sitecore 8.1.
The records in this table are deleted.
- The `DeployDate`, `Status`, and `TaskId` columns are added to the `TreeDefinitions` table.
`NULL` is assigned as the default value for `DeployData` and `TaskId` in all the existing records, `0` is assigned as the default value for `Status` in all the existing records. While these default values should be correct for most of the records, it is better to rebuild the *reporting* database to fill this table with accurate data.
- The `Version` column is added to the `Trees` table.
This is a `ROWVERSION` type column and the values of this column are therefore automatically generated.
- The `Fact_PageViewsByLanguage` and `Fact_RulesExposure` tables are added. These tables are used by the *Content Testing* functionality.
 - The `Fact_RulesExposure` table will never be filled with completely accurate data, because it is one of the newest features in Sitecore 8.1 and the previous interactions do not contain all the information to fill that table. As a result, rebuilding the *reporting* database will not fill this table.
 - The `Fact_PageViewsByLanguage` table could be filled with some data after the upgrade.

To have accurate data in the *reporting* database, you must rebuild it first. Instructions for rebuilding the reporting database can be found at:

<https://doc.sitecore.net/sitecore%20experience%20platform/xdm%20configuration/walkthrough%20rebuilding%20the%20reporting%20database>

Note

Even though this process is resource consuming and may take a considerable amount of time, it is the recommended way to upgrade the *reporting* database from Sitecore 8.0 to Sitecore 8.1.

Publish the site

Publish the entire site (including system items, templates and so on) to every publishing target.

To publish the entire site:

1. On the Sitecore Desktop, click the **Sitecore Start** button and then click **Publish Site**.
2. In the **Publish Site** dialog box, select the **Republish** option.

1.3.4 Upgrading Multiple Instances

Repeat all the steps in the *Upgrade Process* (this chapter) for every Sitecore instance that you would like to update.