

Update Guide for Sitecore Experience Platform

Sitecore XP 10.2.X

How to update your Sitecore XP solution to the latest update release

Introduction

Since the release of Sitecore XP 10.1.0, Sitecore update releases do not contain any breaking changes or database changes. The upgrade process from an earlier version in the Sitecore XP 10.2 series to Sitecore XP 10.2.X has therefore become much easier and all the required steps are described in this guide.

For security reasons, we strongly recommend that you always use the latest version of *Sitecore Identity Server*, which is installed by default during upgrade of Sitecore XP.

If you don't plan to upgrade Sitecore XP, you can upgrade Sitecore Identity only.

Visit the [Sitecore Identity](#) page to find downloads for the latest version with installation and upgrade guides.

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

Containers

If you are using Sitecore containers and want to update to Sitecore XP 10.2.X, follow the steps from the [Upgrade Container Deployment Guide](#).

This guide also contains instructions for updating the Identity Server.

Prerequisites

You must ensure that you have all the software prerequisites for the version of Sitecore that you are updating to.

For more information about the software prerequisites, see the [Sitecore XP Installation Guide for the XP Scaled Topology](#).

On-prem and PaaS solutions

If you are using Sitecore on-prem or an Azure PaaS solution and want to update to Sitecore XP 10.2.X, follow the steps that are appropriate for your current version of Sitecore XP.

Updating from an earlier version in the Sitecore XP 10.2 series

To update from an earlier version in the Sitecore XP 10.2 series to Sitecore XP 10.2.X, follow the steps described in this guide.

Upgrading from Sitecore XP 8.1.0 - 10.1 series

If you are upgrading from Sitecore XP 8.1.0 – 10.1 series, you can upgrade directly to Sitecore XP 10.2.X.

On this upgrade path, you must:

- Follow the instructions in the [Sitecore XP 10.2.0 upgrade guide](#).
- Download and use the artifacts from the [Sitecore XP 10.2.X download page](#).

Upgrading from an earlier version

If your solution is based on an earlier version of Sitecore XP, you must first upgrade to Sitecore XP 8.1.0 before you can upgrade to Sitecore XP 10.2.X. For more information about upgrading to Sitecore XP 8.1.0, see the [Upgrade Guide for Sitecore XP 8.1.0](#).

Update Sitecore XP

Before you update your solution, you should read the release notes for all the releases that came between the version you are upgrading from and Sitecore XP 10.2.X.

To update to Sitecore XP 10.2.X, you must be a local administrator on SQL Server and follow this methodology:

- Install the new version of Sitecore and any modules that you are using beside your existing Sitecore XP installation.
- Connect the new installation to your databases.
- Connect the new installation to your Solr indexes.
- Import the client languages.
- Deploy your solution on top of the new installation.
- Perform the post-upgrade steps (optional).

Install a new instance of Sitecore XP 10.2.X with modules

To install Sitecore XP 10.2.X, follow the instructions in the [Installation Guide](#).

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

Upgrade the Identity Server

The latest version of Sitecore Identity server is installed with Sitecore XP 10.2.X.

If you are not using Sitecore Identity server, [disable](#) it.

To complete the upgrade of Sitecore Identity Server, please refer to [Sitecore Identity Server Upgrade](#) guide.

Connect the databases

When the installation is finished, modify the `ConnectionStrings.config` files to connect the new Sitecore XP 10.2.X instance to the databases from your previous Sitecore XP installation.

You must configure each server role appropriately.

If you are using Sitecore Identity server, modify the `Config\production\Sitecore.IdentityServer.Host.xml` file to point the connection string to the *core* database, or the *security* database if you store the users separately, and restart the Sitecore Identity server to apply the changes.

Connect the Solr indexes

To avoid having to rebuild the indexes, we recommend that you connect the new Sitecore XP 10.2.X instance to the Solr indexes from your previous Sitecore XP installation.

To connect to the Solr indexes, update the appropriate connection strings for [content search](#) and for the [xDB](#) indexes. If you have customized the index names – apply patches for the appropriate [configuration files](#).

You can of course decide to use the new Solr indexes that are installed with Sitecore XP 10.2.X and rebuild all the indexes, as described in the post-update steps.

Import client languages

If you had been using a client language other than English before the update, you must download and import the [latest client translations](#) according to the instructions in the [Sitecore Experience Platform Client Translations Manual](#).

Deploy your Sitecore solution

You should now have the new Sitecore XP 10.2.X instance installed and connected to your databases.

Depending on your development process, you should:

- In your solution, reference the new Sitecore XP 10.2.X components and assemblies.
- Ensure that any configuration changes you have implemented are reconfigured in separate configuration patch files.
- Edit the `Web.config` file and any other non-patchable configuration files to meet your needs.
- Deploy your Sitecore solution on the new Sitecore XP 10.2.X installation.

Post-update steps

Note

If you are updating from the Sitecore XP 10.2 series, all the post-update steps are optional.

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

General maintenance

When you have finished the update and updated any Sitecore modules that you use, and before you start using Sitecore, you can:

- Clear the browser cache
- Republish your website to every publishing target
- [Rebuild the content search indexes](#)
- [Rebuild the link databases](#)

xDB maintenance

If you are using xDB, you can:

- [Rebuild the xDB index](#)
- [Redeploy all the marketing definitions](#)

Hotfixes

The Sitecore XP update release 10.2.1 includes a roll-up of previous hotfixes. Please apply these instructions to all [core roles](#) unless stated otherwise.

As part of the fix for 537827 'When Sitecore resolves a target site, it uses site rootPath and language, database is ignored'

Add the following setting to the Sitecore configuration:

```
<!-- RENDERING - SITE RESOLVING MATCH DATABASE
      Affects how links are rendered when Rendering.SiteResolving is
enabled.
      If true, the link provider will check if the target item
database name matches a site database attribute.
      This ensures that the correct site is resolved when sites have
different databases but the same rootPath and language.
-->
<setting name="Rendering.SiteResolvingMatchDatabase" value="true"/>
```

As part of the fix for 447307 'Sitecore connects to the old SolrCloud server after a warm Disaster Recovery has been triggered'

1. Open the website folder `/App_Config/Sitecore/ContentSearch`.
2. Create a new configuration file `Sitecore.ContentSearch.Solr.EnableConnectionLeaseTimeout.config`.
3. Add the following configuration and save the file:

```
<configuration xmlns:patch="http://www.sitecore.net/xmlconfig/"
xmlns:role="http://www.sitecore.net/xmlconfig/role/"
xmlns:search="http://www.sitecore.net/xmlconfig/search/">
  <sitecore role:require="Standalone or ContentManagement or
ContentDelivery" search:require="solr">
    <contentSearch>
      <indexConfigurations>
        <solrHttpRequestFactory
type="Sitecore.ContentSearch.SolrProvider.SolrNetIntegration.Requests.SolrH
ttpWebRequestAdaptingFactory, Sitecore.ContentSearch.SolrProvider">
          <param
type="Sitecore.ContentSearch.Remote.Http.CompositeHttpRequestFactory,
Sitecore.ContentSearch">
            <initializers hint="list:AddInitializer">
              <!-- Add ConnectionLeaseInitializer to enable
setting the value of ConnectionLeaseTimeout in HttpRequest
              This is useful for disaster recovery scenarios
when hosting multiple solr servers behind a DNS
              -->
                <connectionLeaseTimeout
type="Sitecore.ContentSearch.SolrProvider.SolrNetIntegration.Requests.SolrH
ttpRequestInitializerFactory, Sitecore.ContentSearch.SolrProvider"
factoryMethod="EnableConnectionLeaseTimeout" />
            </initializers>
```

```

        </param>
      </solrHttpRequestFactory>
    </indexConfigurations>
  </contentSearch>
</settings>
  <!-- CONNECTION LEASE TIMEOUT
    The timeout interval for the Solr server HttpRequest
connection lease.
    A value of 00:00:00 means that the active
HttpRequest will immediately be closed after it completes servicing the
request.
    A value of 00:03:00 means that the active
HttpRequest will be closed 3 minutes after it completes servicing the
request.
    Default value: 00:03:00
  -->
  <setting
name="ContentSearch.Solr.HttpRequest.ConnectionLeaseTimeout"
value="00:03:00" />
</settings>
</sitecore>
</configuration>

```

As part of the fix for 561884 '[Solr] Use Delete-By-Id instead of Delete-By-Query for deleting index documents'

Add the following setting to the Sitecore configuration:

```

<!-- RESTRICT DELETE BY INDEX NAME
    If enabled, the delete command removes only documents in the solr
core/collection which are associated with the current index.
    The value must be set to true, if multiple indexes are stored in a
single Solr core/collection.
    Default vaue: true
  -->
  <setting name="ContentSearch.Solr.RestrictDeleteByIndexName" value="true"
/>

```

Note

This setting can be set to false only if ALL Sitecore indexes are stored in dedicated Solr cores/collections.

As part of the fix for 97020 '[Solr] Implement soft commits'

Add the following setting to the Sitecore configuration:

```

<!-- USE SOFT COMMITS
    If enabled, soft commits are used.
    The auto-commit must be enabled and configured in solr before
enabling this setting.
    Default vaue: false
  -->
  <setting name="ContentSearch.Solr.UseSoftCommits" value="false" />

```

Note

If the setting is enabled, the auto-commit must be enabled and configured in Solr.

As part of the fix for 493002 'Permission Enabled Search Does Not Work in Nested Role Scenario'

1. Add the following setting to the Sitecore configuration:

```
<!-- CONTENT SEARCH - INDEXING READACCESS WITH PRIORITY TO ENABLE FIX FOR
493002
    This setting specifies whether or not to indexing readaccess
    permission with proprity and search with prioriry.
    When enabled it will have extra priority to the readaccess
    permission when indexing e.g. User > Entity > Descendents.
    Default value: false
-->
<setting name="ContentSearch.Enable-493002" value="false"/>
```

2. Rebuild indexes.

As part of the fix for 568955 'Item version number renumbered after installing Sitecore Package'

Add the following setting to the Sitecore configuration:

```
<!-- VERSION INSTALL MODE FOR NEW ITEMS
    Sets the default installation mode for the versions of new items
    when installing a Sitecore package.
    Merge - merge versions by version number
    Append - append version if version with same revision number
    does not exist
    Default value: Append
-->
<setting name="VersionInstallModeForNewItems" value="Append"/>
```

As part of the fix for 576660 'Security enhancement'

1. Remove `sitecore_xaml.ashx` handler from `web.config` on ContentDelivery instances.

2. Remove the following lines from `sitecore.config` on ContentDelivery instances:

```
<handler trigger="-/xaml/" handler="sitecore_xaml.ashx"/>
<handler trigger="~/xaml/" handler="sitecore_xaml.ashx"/>
```