Express Migration Tool Guide

Sitecore XP 8.2

How to upgrade from Sitecore 7.2 directly to Sitecore 8.2

sitecore®
Own the experience™

Table of Contents

Chapter 1

Introduction

This document describes how to use and execute the Express Migration Tool rev. 160811.

This document contains the following chapters:

- **Chapter 1 – Introduction**
  An introduction to the Express Migration Tool installation process.
- **Chapter 2 – Preprocessing**
  An outline of the requirements and necessary steps you must take before the migration.
- **Chapter 3 – The Migration Process**
  Describes all the steps of the migration process, the result, and errors.
- **Chapter 4 – Post-migration steps**
  Step-by-step instructions for completing the migration process.

1.1 Getting started

The Sitecore Express Migration Tool is designed to help you migrate directly from any version of Sitecore 7.2 to the latest version of Sitecore 8.2. The Sitecore instance that you are migrating to is called the target instance.

The Express Migration Tool:

- Migrates one Sitecore instance at a time.
- Compares your Sitecore instance with a clean version of the same product.
- Copies all of the items and files that have been changed to the target instance.

1.1.1 Supported versions

The Sitecore Express Migration Tool lets you migrate from Sitecore 7.2 to Sitecore Experience Platform™ 8.2 initial release.

You can upgrade from the following versions of Sitecore 7.2:

- Sitecore 7.2 rev.160123 (7.2 Update-6)
- Sitecore 7.2 rev.151021 (7.2 Service Pack-2, originally 7.2 Update-5)
- Sitecore 7.2 rev.150408 (7.2 Update-4)
- Sitecore 7.2 rev.141226 (7.2 Service Pack-1, originally 7.2 Update-3)
- Sitecore 7.2 rev.140526 (7.2 Update-2)
- Sitecore 7.2 rev.140314 (7.2 Update-1)
- Sitecore 7.2 rev.140228 (7.2 Initial Release)

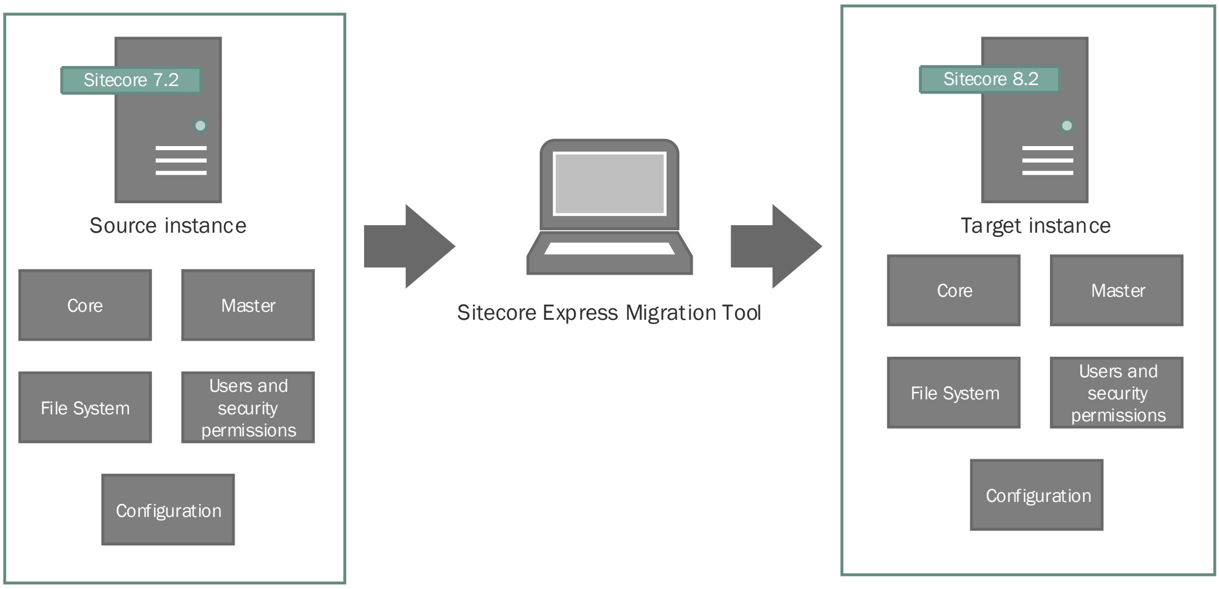
Note
The Express Migration Tool only supports the migration of a SQL-based setup of the source and target instances. Oracle database setup is not supported in this version of the tool.

1.2 A generic setup

The Sitecore Express Migration Tool copies items and files from one Sitecore instance at a time.

The tool supports the migration of remote servers.

In a generic setup, the source instance, the target instance, and the Sitecore Express Migration Tool can be on three different computers.



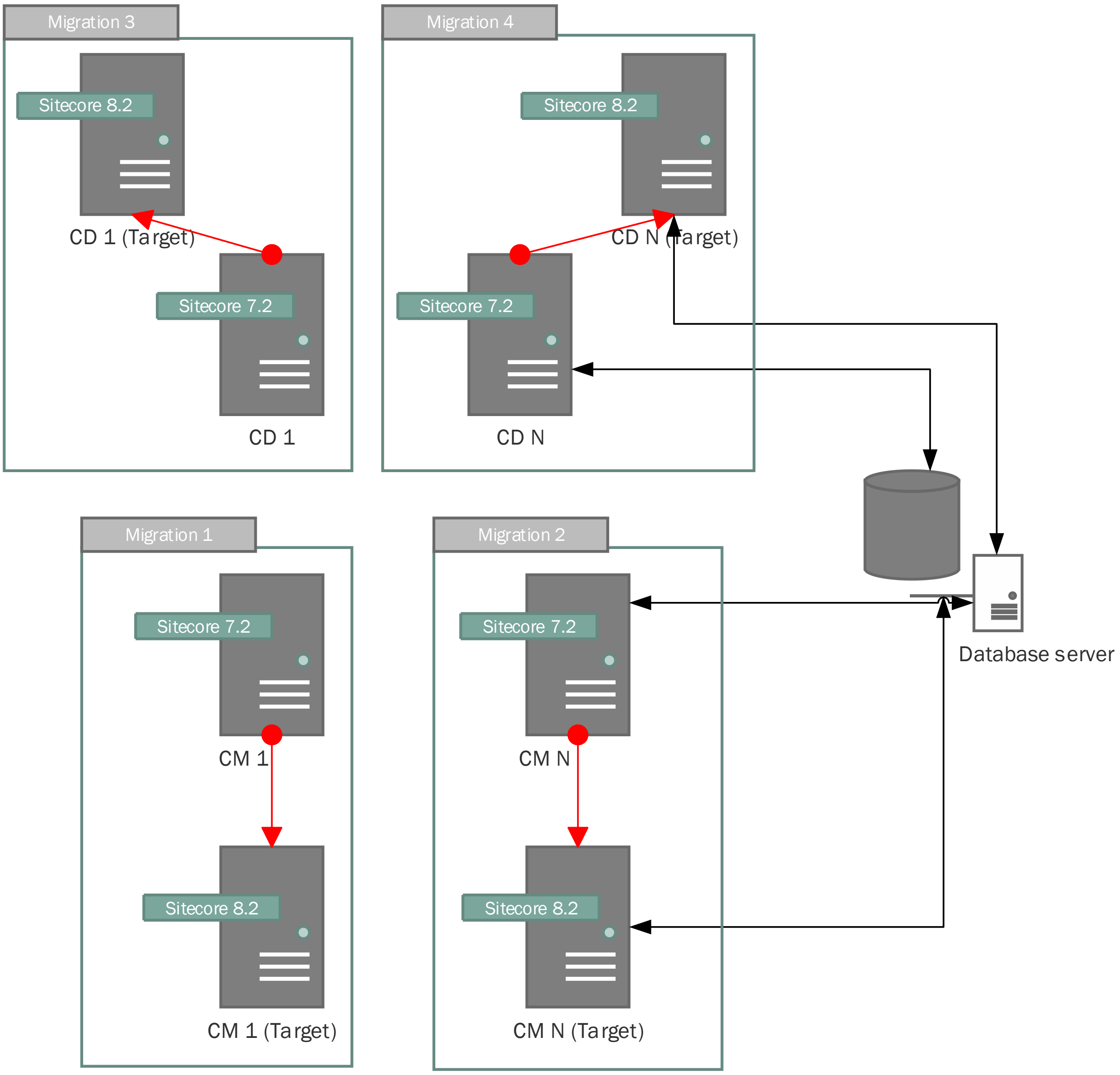
1.2.1 A distributed environment

In a scaled setup, you must migrate each Sitecore instance separately. The strategy that you use to migrate a distributed environment can vary, depending on the Sitecore solution you have.

One instance of the Sitecore Express Migration Tool can migrate one instance of Sitecore at a time. Therefore, in a scaled environment and depending on your migration strategy, you can run the tool once for each of the servers. The specific settings can vary from server to server.

For example, if you want to migrate four servers, you can run the tool four times, and run a new instance of the tool for each server you are migrating. In a different scenario, however, you can run the tool only once and configure the remaining three servers.

The following example shows the migration of four servers in a distributed environment:



Migration 1

Use the Sitecore Express Migration Tool to migrate the content management server (CM 1).

Migration 2

Use the Sitecore Express Migration Tool to migrate the configuration and file system of the other content management server (CM N). You do not have to migrate the Core database, the Master database, and the user accounts on this server because it uses the same databases as CM 1.

Migration 3 and 4

Set up and configure the new content delivery servers. You can use the Express Migration Tool to help you migrate the configuration.

To move the data from the CM servers, you must modify the connection strings to point to the migrated databases.

Chapter 2

Preprocessing

This chapter describes how to prepare for the migration, the system requirements, and the installation of a comparison instance.

This chapter contains the following sections:

- Preprocessing
- Configuring and running

2.1 Preprocessing

Before migrating any Sitecore instances:

- Ensure that your hardware and software meet the requirements for a Sitecore installation and for the Express Migration Tool.
- Ensure that all scheduled tasks are finished and that the event queue has been processed.
- Stop active A/B or MV tests.
- Enforce a content freeze on your system.
- Make a backup of your source instance.

  Although the migration tool does not modify the source instance, you should back up your servers before running the Express Migration Tool.
- Download and install the latest version of Sitecore Experience Platform™ 8.2 from the Sitecore Developer Portal – https://dev.sitecore.net

  This is your target instance.
- If you are comparing your source instance with a comparison instance, you must install the comparison instance first.

> **Note**
> When you migrate a Sitecore instance, the Express Migration Tool interacts with the database. This can degrade the performance of the database server and the performance of your website.
>
> This can also slow down the migration process. To avoid this, you can use a separate database server or ensure that your content delivery server has enough hardware resources.

2.1.1 Operating system requirements

The Express Migration Tool is only compatible with the 64-bit version of Windows operating systems. However, the source and target instances can be run on a 32-bit version of Windows.

The Express Migration Tool can be run on the following operating systems:

- Windows Server 2008 (64-bit)
- Windows Server 2008 R2 (64-bit)
- Windows Server 2012 (64-bit)
- Windows 2012 R2 (64-bit)
- Windows 7 (64-bit)
- Windows 8 (64-bit)
- Windows 8.1 (64-bit)
- Windows 10 (64-bit)

2.1.2 .NET Framework requirements

The Express Migration Tool requires .NET Framework 4.5.2 or higher.

You must apply any available updates to the .NET Framework to every Sitecore instance.

2.1.3 Hardware requirements

The Express Migration Tool has the following minimum hardware requirements:

- Processor: Intel Pentium 4, 2 GHz or faster.
- RAM: 4GB minimum, 8GB – recommended.
- Storage: Ensure your target instance has at least the same amount of free hard disk space as your source instance.

  Moreover, because the Express Migration Tool keeps the analysis information about instances and states in the local database, it needs at least an additional 4GB of free hard disk capacity.

2.2 Configuring and running

Before you run the Express Migration Tool, there are some optional configuration settings that you might want to configure.

The optional settings enable you to configure the maximum number of retries and interval between retries after an error, as well as which files and folders the tool should ignore during the migration process.

2.2.1 Configuring the Express Migration Tool

If the connection to a database is lost during migration, the tool automatically retries to send the database requests that failed. You can configure the intervals in miliseconds between every retry (the default value is 3,000 milliseconds), and the maximum number of retries that the tool should perform (the default value is five retries).

You can also configure which files and folders the tool should ignore during analysis and migration. This is useful if you have a folder with cached media, a log folder, or temporary files, and you do not want to migrate the folder to the target instance. When you run the tool, the ignored files and folders do not appear on the file selection step.

To configure the interval between retries and the maximum number of retries:

1. In a text editor, edit the `App_Config\Default.config` file.
2. Find the following setting and specify the value:
   - Interval retries:

   `<setting name="Database.Retry.Interval" value="3000" />`

   - Maximum number of retries:

   `<setting name="Database.Retry.MaxTries" value="5" />`
3. Save your changes.

To configure the files and folders that the tool should ignore:

1. In a text editor, edit the `App_Config\Default.config` file.
2. In the `<filterConfigurator>` section, find the `<directory hint="list:IgnoreDirectory">` node and configure the files and the folders that the tool should ignore during migration. For example:

```
       <filterConfigurator
type="Sitecore.ExpressMigration.Core.Files.Filters.FileFilterConfigurator,
Sitecore.ExpressMigration.Core">
                    <directory hint="list:IgnoreDirectory">
                      <configs>App_Config</configs>
                      <dll>bin</dll>
                      <temp>temp</temp>
```

3. Save your changes.

2.2.2 Running the Express Migration Tool

You can download the Express Migration Tool from the Sitecore Developer Portal.

1. Unzip the file.
2. Navigate to the path where the tool is installed, and launch the `Sitecore.ExpressMigration.exe` application.

Chapter 3

The Migration Process

This chapter describes how to use and configure the tool. It explains the migration process in detail and describes how to migrate a standard Sitecore instance, as well as how to migrate an instance that also contains some other Sitecore modules.

This chapter contains the following sections:

- Using the Express Migration Tool
- Migration result
- Installing and using a comparison instance
- Known issues

3.1 Using the Express Migration Tool

Before you run the Express Migration Tool, you must decide whether you want to run it on the same computer as your source and target instances or on a different computer.

If you run the tool on a different computer, you must ensure that it has Server Message Block (SMB) access to the source, target, and comparison instance's database servers and file system, and then you can specify the network path to the file system on the source instance and the comparison instance.

You can run the tool in the following scenarios:

- The Express Migration Tool is run on the same computer where the source instance is installed, while the target instance is located on a remote server.
- The Express Migration Tool and target instance are run on the same computer, and the source instance is located on a remote server.
- The Express Migration Tool, source, and target instances are located on separate computers.
- The Express Migration Tool, source, and target instances are located on the same computer.

3.1.1 Running the Express Migration Tool

The Express Migration Tool contains four progressive tabs:

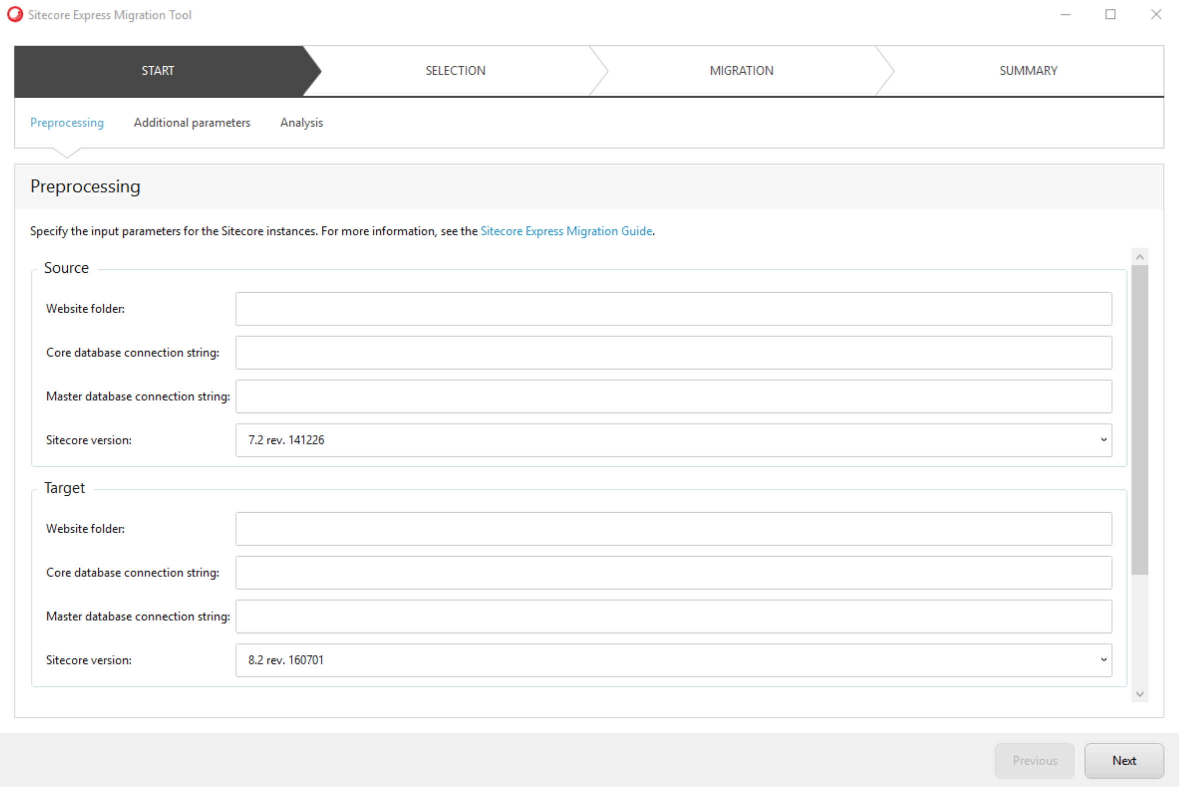
- Start
- Selection
- Migration
- Summary

On the **Start** progressive tab, you must provide the tool with the relevant parameters of your source and target instances. Then, you can select the parts of the content that you want to migrate.

After the analysis has finished, the Express Migration Tool presents you with lists of the items and files in the Core and Master databases that you can migrate.

When you have installed and configured the tool, you are ready to begin the migration process:

1. Navigate to the path where you downloaded the tool, and launch the `SitecoreExpressUpgrades.exe` application. The tool displays the following window:



Preprocessing

Specify the input parameters for the Sitecore instances. For more information, see the Sitecore Express Migration Guide.

Source

Website folder:

Core database connection string:

Master database connection string:

Sitecore version: 7.2 rev. 141226

Target

Website folder:

Core database connection string:

Master database connection string:

Sitecore version: 8.2 rev. 160701

2. On the **Start** progressive tab, on the **Preprocessing** subtab, in the **Source** section, in the **Website folder** field, enter the path to your source instance.

3. In the **Core database connection string** field, enter the connection string for the Core database on your source instance.

   The connection strings are stored in the `\Website\App_Config` folder of the relevant Sitecore source or target instance, in the `ConnectionStrings.config` file.

   Depending on how you set up your instances, you can modify the connection strings when you enter them in the tool.

   For example:

```
Data Source=(local);Initial Catalog=sc72rev140526Sitecore_core;Integrated
Security=False;User ID=sa;Password=Password1234
```

Note
Connection strings can have different formats. For example, you can have a connection string that does not contain the `User ID` and `Password` attributes, but instead contains the `Integrated Security=True` attribute.

4. In the **Master database connection string** field, enter the connection string for the Master database on your source instance.

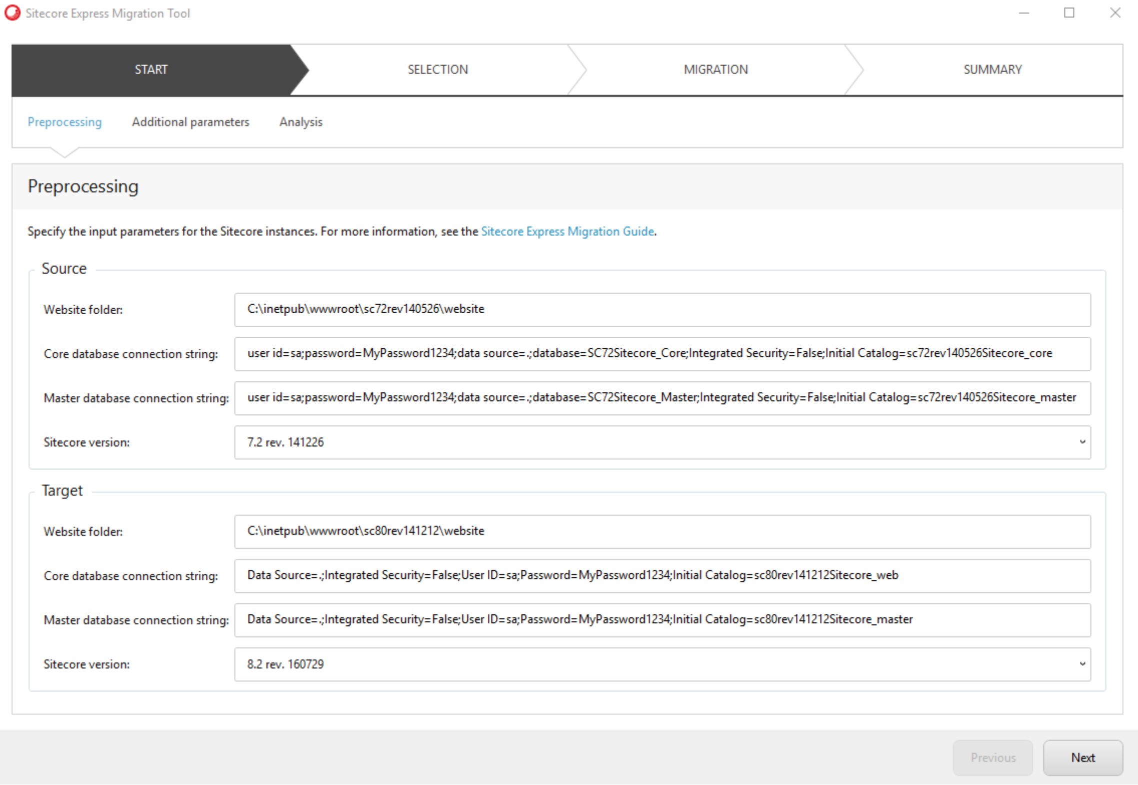
   If you have a source or target instance on a different computer than the Express Migration Tool, you must ensure that the databases are accessible and the connection strings are valid for the computer you are running the tool from.

You can specify a non-local connection string for both the **Core database connection string** field, or the **Master database connection string** field. To do this, in the `Data Source` parameter, enter the network name of the server.

For example:

```
Data Source=ProdServer;Initial Catalog=sc81rev160302Sitecore_core;Integrated
Security=False;User ID=sa;Password=Password1234
```

5. In the **Sitecore version** field, select the version of your source instance from the drop-down list.
6. Repeat this process for the **Target** section and when you are finished, click **Next**.



Preprocessing

Specify the input parameters for the Sitecore instances. For more information, see the Sitecore Express Migration Guide.

Source

Website folder: C:\inetpub\wwwroot\sc72rev140526\website

Core database connection string: user id=sa;password=MyPassword1234;data source=.;database=SC72Sitecore_Core;Integrated Security=False;Initial Catalog=sc72rev140526Sitecore_core

Master database connection string: user id=sa;password=MyPassword1234;data source=.;database=SC72Sitecore_Master;Integrated Security=False;Initial Catalog=sc72rev140526Sitecore_master

Sitecore version: 7.2 rev. 141226

Target

Website folder: C:\inetpub\wwwroot\sc80rev141212\website

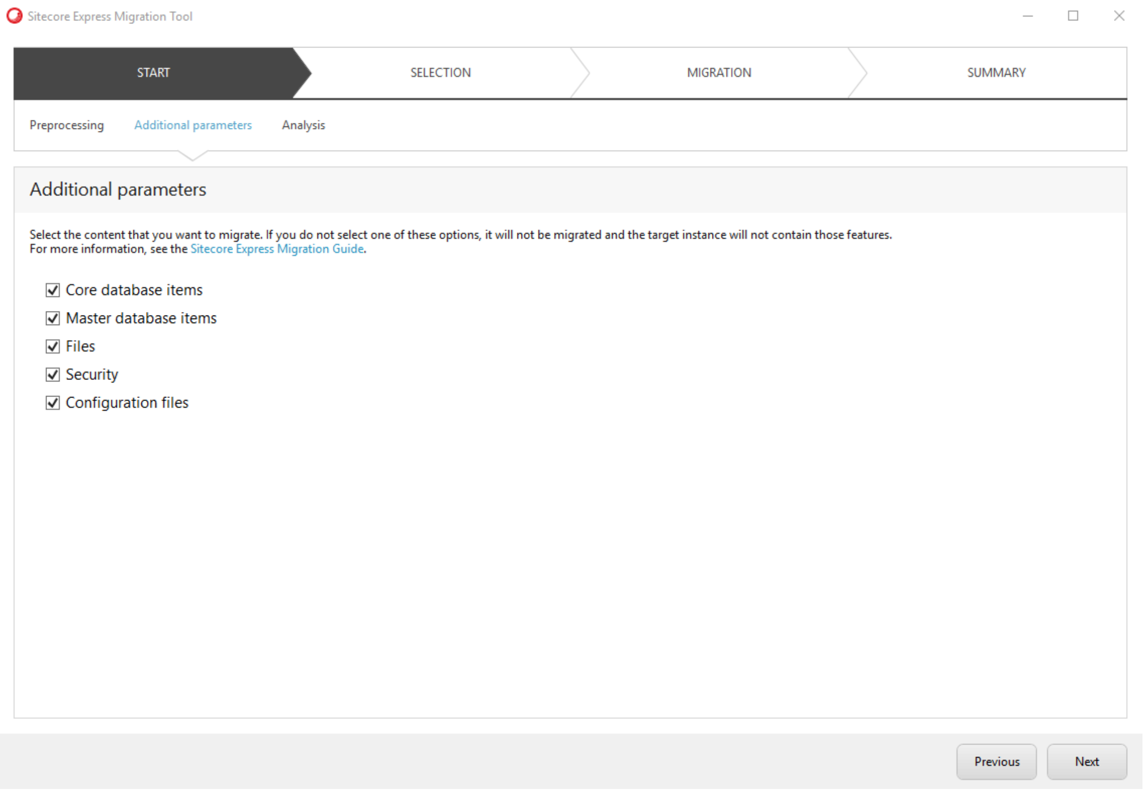
Core database connection string: Data Source=.;Integrated Security=False;User ID=sa;Password=MyPassword1234;Initial Catalog=sc80rev141212Sitecore_web

Master database connection string: Data Source=.;Integrated Security=False;User ID=sa;Password=MyPassword1234;Initial Catalog=sc80rev141212Sitecore_master

Sitecore version: 8.2 rev. 160729

7. On the **Start** progressive tab, on the **Additional parameters** subtab, select the options you want to migrate.

Important
When you migrate an instance, you must select all the options that you want to migrate, and not attempt to perform an incremental migration.



Additional parameters

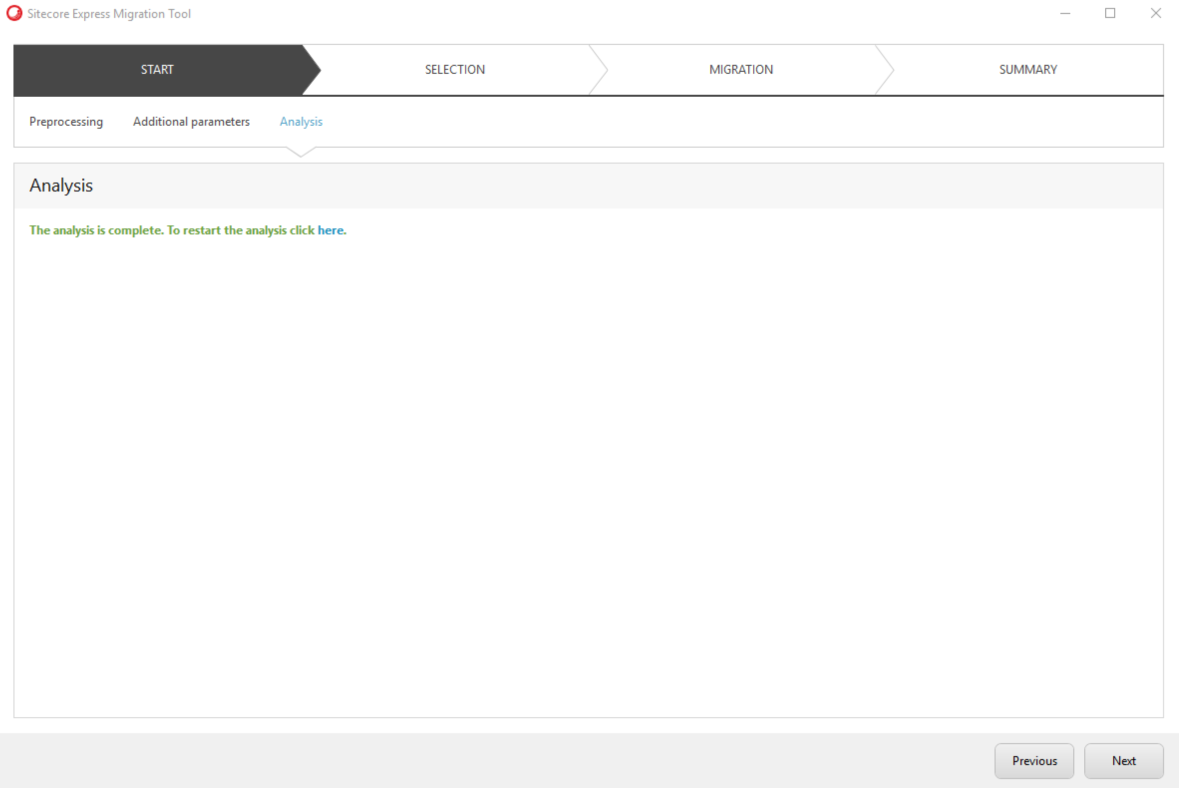
Select the content that you want to migrate. If you do not select one of these options, it will not be migrated and the target instance will not contain those features.
For more information, see the Sitecore Express Migration Guide.

☑ Core database items
☑ Master database items
☑ Files
☑ Security
☑ Configuration files

8. Click **Next**.

   The tool begins the analysis of the source instance to identify all the new and modified files, items, and so on.

If there is an error during the analysis or if it was not completed successfully, you can restart the analysis.



The analysis is complete. To restart the analysis click here.

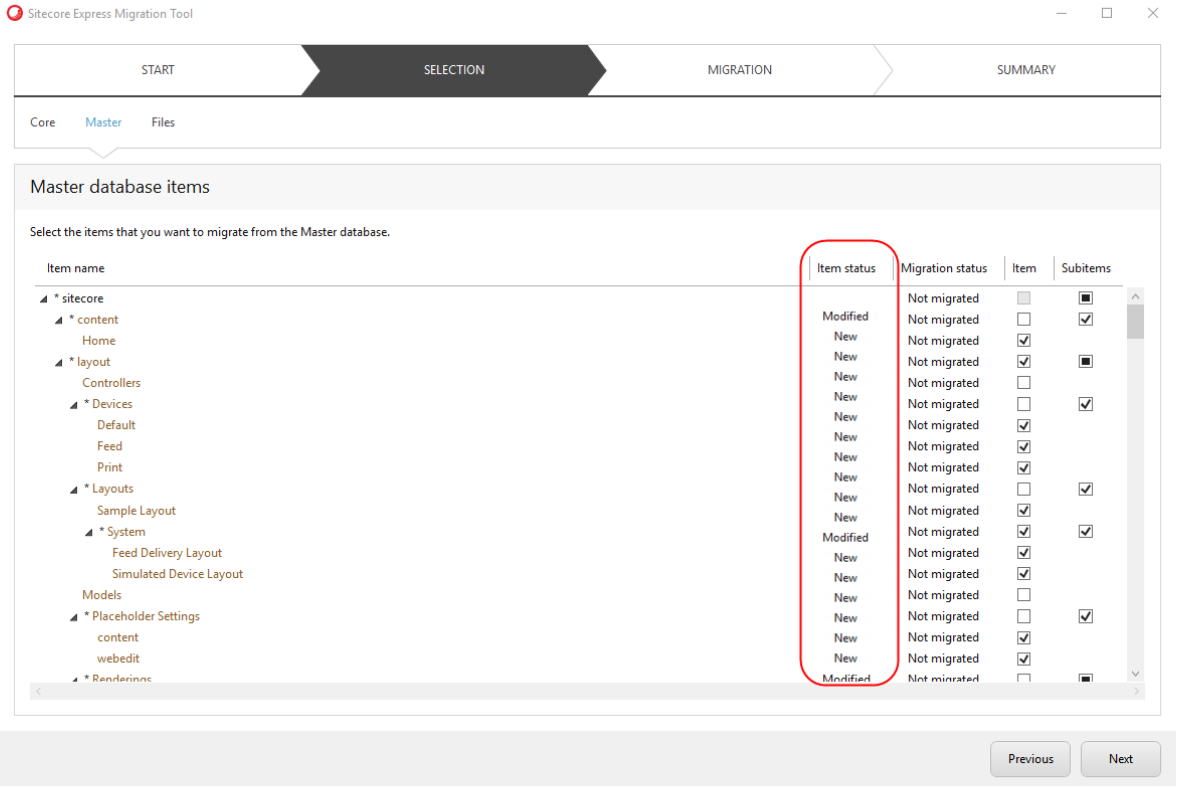
Selecting the items to migrate

When the tool finishes analyzing the Sitecore instances that you specified, it opens the **Selection** progressive tab.

The **Selection** progressive tab contains three subtabs:

- Core
- Master
- Files

The subtabs lists all the items and files in the Core and Master databases in the source instance, as well as all the system files.



Master database items

Select the items that you want to migrate from the Master database.

Item name	Item status	Migration status	Item	Subitems
◢ * sitecore		Not migrated	□	■
◢ * content	Modified	Not migrated	□	☑
Home	New	Not migrated	☑	
◢ * layout	New	Not migrated	☑	■
Controllers	New	Not migrated	□	
◢ * Devices	New	Not migrated	□	☑
Default	New	Not migrated	☑	
Feed	New	Not migrated	☑	
Print	New	Not migrated	☑	
◢ * Layouts	New	Not migrated	□	☑
Sample Layout	New	Not migrated	☑	
◢ * System	Modified	Not migrated	☑	☑
Feed Delivery Layout	New	Not migrated	☑	
Simulated Device Layout	New	Not migrated	☑	
Models	New	Not migrated	□	
◢ * Placeholder Settings	New	Not migrated	□	☑
content	New	Not migrated	☑	
webedit	New	Not migrated	☑	
◢ * Renderings	Modified	Not migrated	□	■

The Express Migration Tool identifies all the files and items that have been added, modified, or deleted on the source instance, and automatically selects content for the migration.

In the **Item Status** column, you can see items and files tagged according to their state:

- Modified
- Deleted
- New

The following specific items and files are preselected for migration:

- All content and content-related items: templates, marketing attributes, and layouts.
- Modified content items.
- All the new files on the source instance.

The following items and files are not preselected for migration:

- All the system items and files.

  Migrating the system items and files can cause breaking changes on the target instance.

  A breaking change occurs when a change is made in the product that causes other components to fail.

  If a file with the same name exists on both the source and target instance, it is overwritten on the target instance with the version from the source instance.

  Therefore, all the Sitecore system items and files are not automatically selected for migration regardless of changes. If you are sure that the changes in a file or item do not cause any problems, then you can manually select the file or item for migration.
- All deleted and modified files, because of the risk of breaking the target instance.

- Modified templates.

During migration, the tool processes the selected items and files:

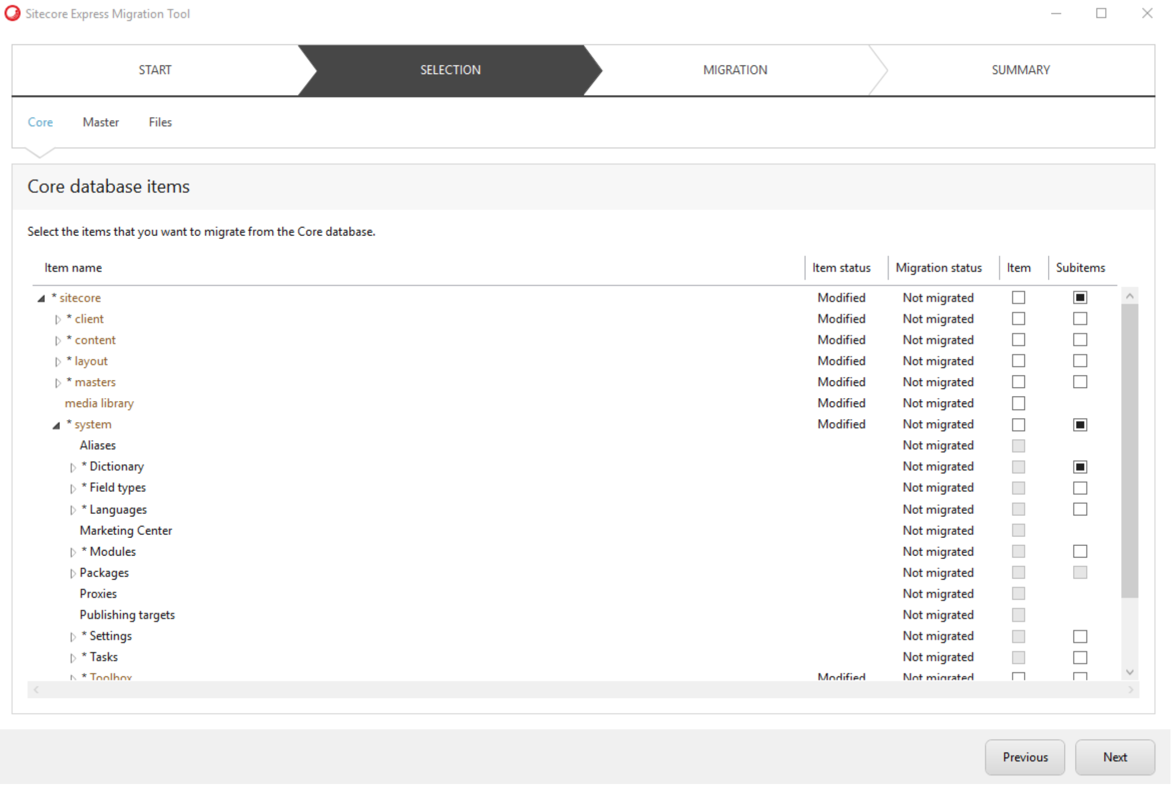
- New items are created in the target database.
- All of the files that are related to the selected media items are copied to the target instance.
- New files are copied to the target file system.
- Modified items are overwritten on the target instance.
- Deleted items and files in the source instance are deleted from the target database or file system.

> **Note**
> The Express Migration Tool preselects modified content items for migration, but does not preselect modified templates.
>
> If you have modified any templates in your source instance and you want to migrate them, you must select them manually. To do this, expand the tree and select the relevant items or files. You can select an individual item or file or the entire subtree.

The Express Migration Tool helps you migrate the configuration files from the source instance. The tool analyzes the configuration files on the source instance and identifies the changes that were made compared to a clean instance of the same version.

To select the items you want to migrate:

1. In the **Core database items** window, select the files and items that you want to migrate. You can also clear the selection of files and items that you do not want to migrate. Click **Next**.



Core database items

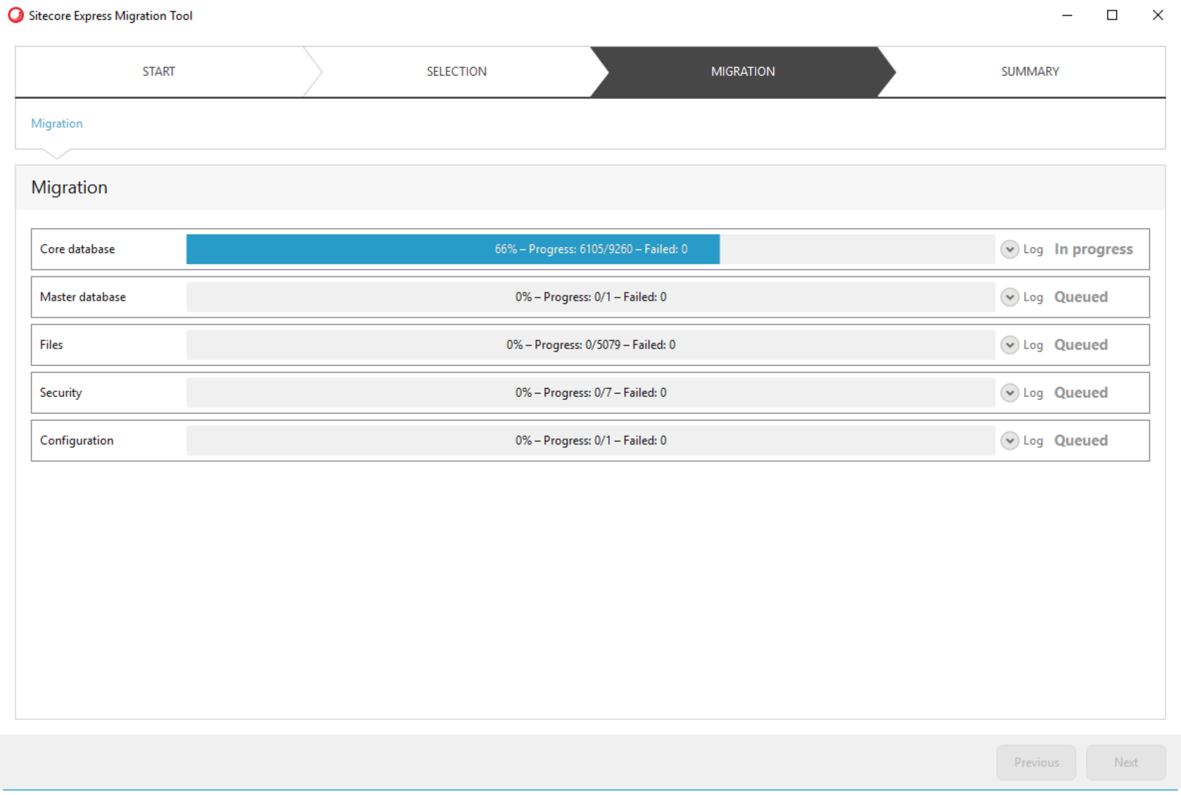
Select the items that you want to migrate from the Core database.

Item name	Item status	Migration status	Item	Subitems
* sitecore	Modified	Not migrated	□	▣
* client	Modified	Not migrated	□	□
* content	Modified	Not migrated	□	□
* layout	Modified	Not migrated	□	□
* masters	Modified	Not migrated	□	□
media library	Modified	Not migrated	□	
* system	Modified	Not migrated	□	▣
Aliases		Not migrated	□	
* Dictionary		Not migrated	□	▣
* Field types		Not migrated	□	□
* Languages		Not migrated	□	□
Marketing Center		Not migrated	□	
* Modules		Not migrated	□	□
Packages		Not migrated	□	□
Proxies		Not migrated	□	
Publishing targets		Not migrated	□	
* Settings		Not migrated	□	□
* Tasks		Not migrated	□	□
* Toolbox	Modified	Not migrated	□	□

> **Note**
> Modified items and subitems in the tree are shown with an asterisk. Unmodified items that contain modified and/ or removed and/or new subitems are also marked with an asterisk.

2. Repeat the process in the **Master database items** window and the **Files** window. Click **Next**.

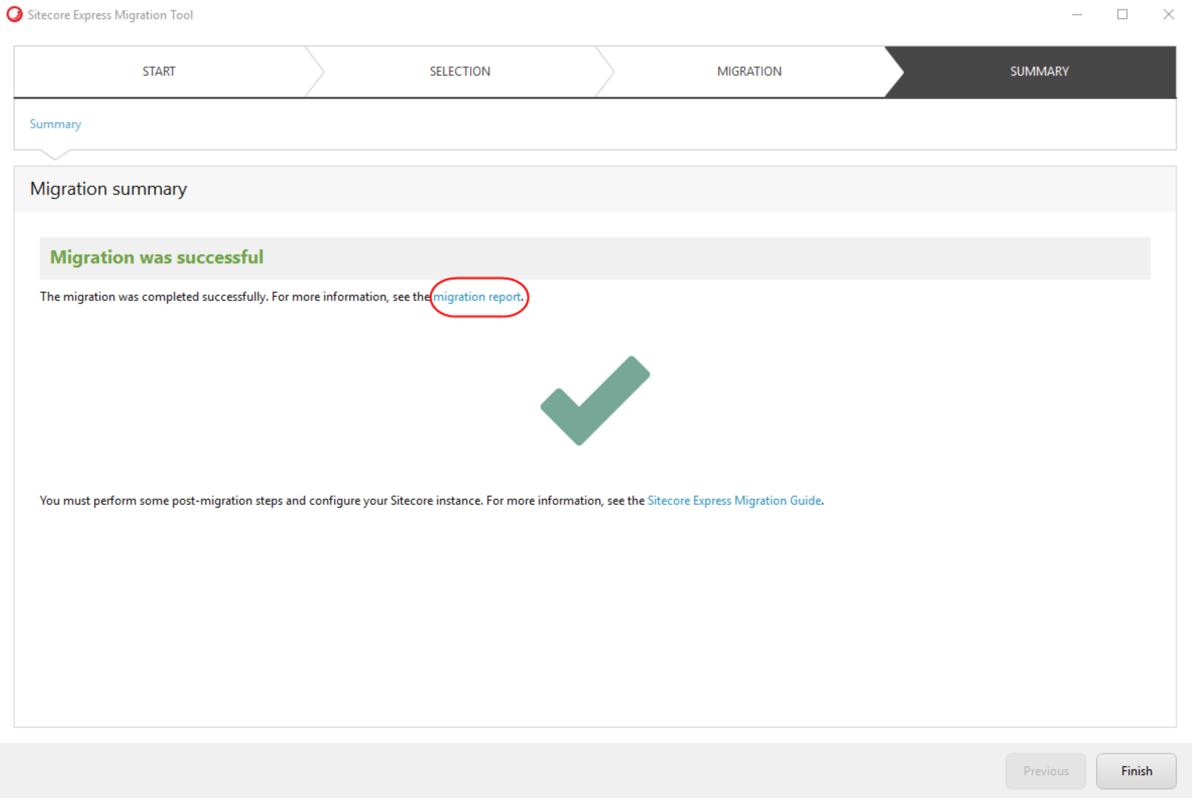
At this point, the Express Migration Tool begins to migrate the files to the new Sitecore instance.



Migration

Core database	66% – Progress: 6105/9260 – Failed: 0	Log	In progress
Master database	0% – Progress: 0/1 – Failed: 0	Log	Queued
Files	0% – Progress: 0/5079 – Failed: 0	Log	Queued
Security	0% – Progress: 0/7 – Failed: 0	Log	Queued
Configuration	0% – Progress: 0/1 – Failed: 0	Log	Queued

3. When the migration is completed successfully, click **Finish** to close the migration wizard, or if you want to open the migration report, click the **migration report** link.



Migration summary

Migration was successful

The migration was completed successfully. For more information, see the migration report.

You must perform some post-migration steps and configure your Sitecore instance. For more information, see the Sitecore Express Migration Guide.

You have now successfully migrated your source instance to the target Sitecore 8.2 instance.

3.2 Migration result

When the migration process is completed, the following objects from the source instance are available on the target instance:

- The items and files in the Core and Master database that you selected for migration.
- The user roles and accounts.
- The configuration files that you selected.

If a migrated item with the same name exists on both the target and the source instances, the tool overwrites the item on the target instance with the item from the source.

Items in the Core and Master Database

The tool migrates all the items that are related to custom content on the source instance by default:

- Items are migrated without their templates.

  The tool does not automatically migrate the template of a selected item. The tool preselects modified content items for migration, but does not preselect out-of-the-box, modified templates.
- For media library assets, the tool also migrates the respective database records.

Security

When migration is complete:

- All the custom user roles and user accounts from the source instance are migrated to the target instance.
- All the security permissions are migrated.
- All the account passwords are migrated, so users can log in to the target system.
- All the user accounts are members of the same security groups as they were on the source instance.

If a role with the same name exists on the source and on the target instances, the resulting role on the target instance has the same permissions and membership as the role on the source instance.

Files

All the files that you selected are migrated to the target instance.

Configuration files

When migration is complete:

- All the new configuration files are migrated without changes to the `<target instance root folder>\Migration.MigratedConfiguration` folder and with the `.disabled` extension added to the filenames.
- The tool generates patch files that highlight the differences between the source instance and the clean instance.

  Sometimes the tool cannot generate configuration file patches, for example, when there are non-Sitecore sections in the `web.config` file and the `App_Config\Security\GlobalRoles.config` file.

  When this happens, the tool generates `diff` files that contain all the configuration nodes that have been changed on the source instance, compared to the clean instance.

  The tool identifies the critical places where known breaking changes have occurred, as well as the nodes that have changed, and adds XML comments to the configuration patch and `diff` files.

However, the tool does not automatically apply the generated patches and `diff` files to the breaking changes between the source and the target instance. Therefore, you must apply patches manually.

3.3 Installing and using a comparison instance

The Express Migration Tool contains metadata about every version of Sitecore 7.2. When you run the tool, you must select the Sitecore version that you are using. Then, the tool compares the source instance with the metadata for the same version and only migrates new or modified content and ignores the items that have not been changed.

However, you may want to compare your Sitecore 7.2 instance with a custom instance. This instance is called a *comparison instance*.

You can use a comparison instance when:

- You have modules installed on your source instance and you want to automatically exclude the migration of their system items and files.

  When you install a module on the comparison instance, the tool marks the items as unchanged and does not copy them to the target instance.

- You want to exclude some patches or hotfixes from the migration process.
- You want to remove some configuration options from the migration process.

To use a comparison instance:

1. Install the appropriate Sitecore instance.

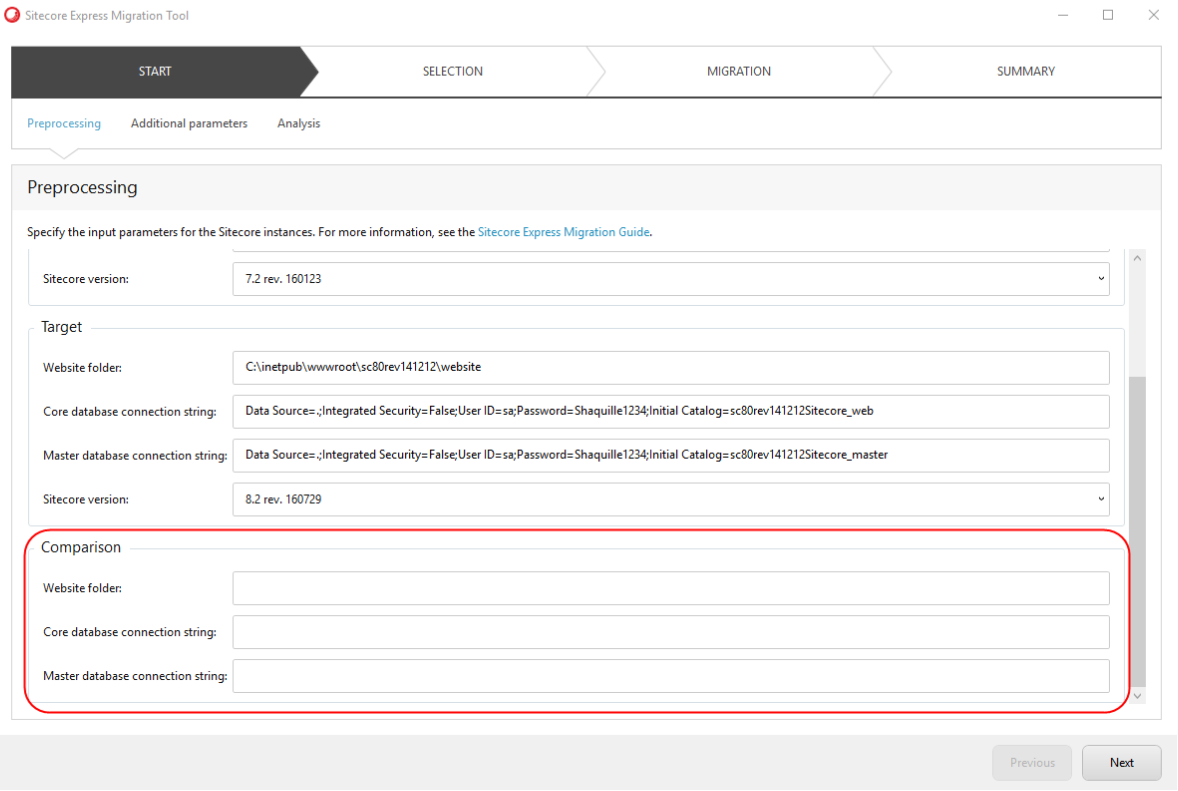
   You must use the same version as the source instance that you want to migrate. You can download it from the Sitecore Developer Network.

2. Configure the comparison instance and install all the required modules, patches, and files.

   Note
   You do not have to deploy the comparison instance.

3. To configure the tool to use a comparison instance, you must disable the `App_Config\Include\Metadata.config` file by adding the `.disabled` extension to the filename, and run the tool.

When you enable the configuration file, on the **Start** progressive tab, in the **Preprocessing** window, the **Comparison** section appears. This enables you to configure the connection strings for the comparison instance.



Preprocessing

Specify the input parameters for the Sitecore instances. For more information, see the Sitecore Express Migration Guide.

Sitecore version: 7.2 rev. 160123

Target

Website folder: C:\inetpub\wwwroot\sc80rev141212\website

Core database connection string: Data Source=.;Integrated Security=False;User ID=sa;Password=Shaquille1234;Initial Catalog=sc80rev141212Sitecore_web

Master database connection string: Data Source=.;Integrated Security=False;User ID=sa;Password=Shaquille1234;Initial Catalog=sc80rev141212Sitecore_master

Sitecore version: 8.2 rev. 160729

Comparison

Website folder:

Core database connection string:

Master database connection string:

4. When you run the tool, in the **Comparison** section, specify the parameters for the comparison instance in the same way as you did with the **Source** and **Target** instance sections. Click **Next**.

   The tool then compares your source instance with the comparison instance instead of comparing it with a clean Sitecore instance.

The migration report

The migration report is an `.html` file that contains detailed statistical information about the migrated instance, settings, and the results of the migration, including the number of items that were successfully migrated and those that failed.

To open the migration report, click the link in the **Migration summary** window, or navigate to the `\Sitecore Express Upgrade Tool\MigrationReports` folder, and open the `MigrationReports*timestamp.html` file.

The log files

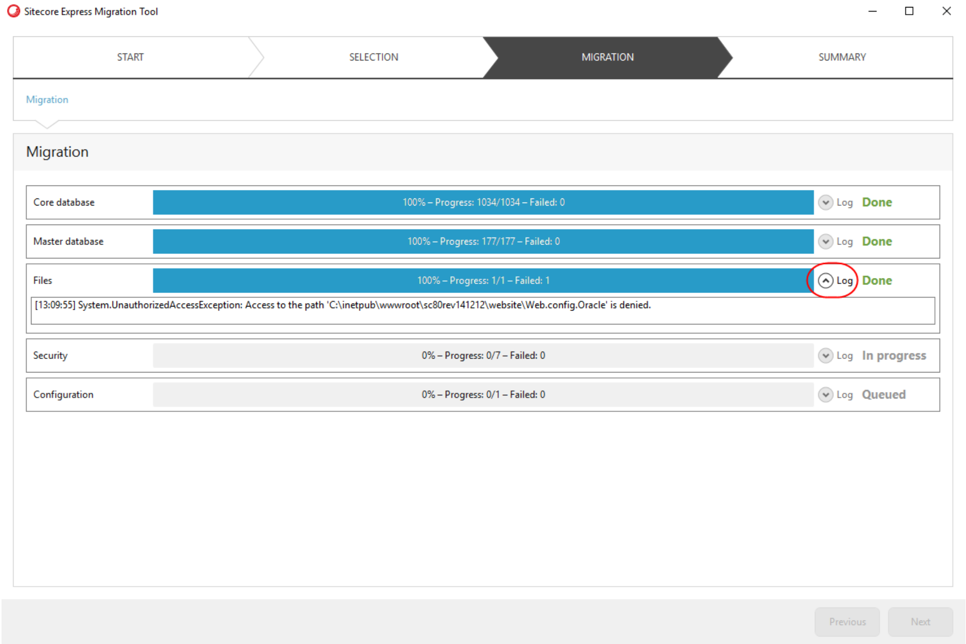
The log files are `.txt` files that keep a record of the changes the Express Migration Tool made during migration.

You can find the log files in the `\Sitecore Express Upgrade Tool\Logs` folder.

3.3.1 Migration errors

When there is an error during the migration process, the tool automatically retries to migrate the files and items that failed. If the tool fails to migrate the items or files again, in the **Migration** window, you can open the log file for that specific migration option and see a list of the error(s) that occurred.

To see the errors, click the **Log** drop-down arrow. The information displayed is the same as the information in The log files.



Core database 100% – Progress: 1034/1034 – Failed: 0 Log Done

Master database 100% – Progress: 177/177 – Failed: 0 Log Done

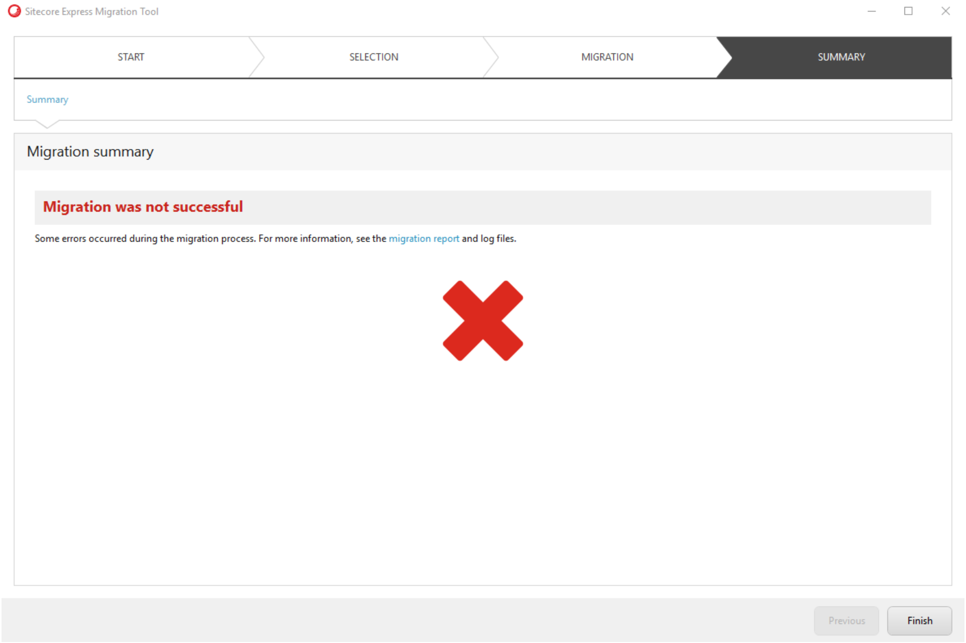
Files 100% – Progress: 1/1 – Failed: 1 Log Done

```
[13:09:55] System.UnauthorizedAccessException: Access to the path 'C:\inetpub\wwwroot\sc80rev141212\website\Web.config.Oracle' is denied.
```

Security 0% – Progress: 0/7 – Failed: 0 Log In progress

Configuration 0% – Progress: 0/1 – Failed: 0 Log Queued

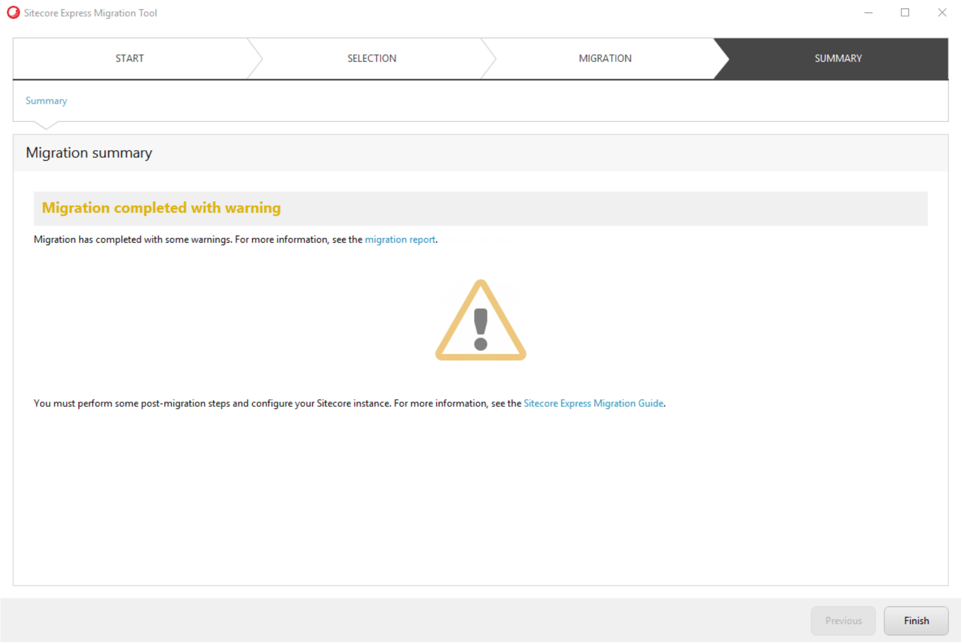
If migration was not successful, on the **Summary** tab, you can open The migration report for more details about the errors.



Migration was not successful

Some errors occurred during the migration process. For more information, see the migration report and log files.

The tool can also finish the migration process with warnings.



Migration summary

Migration completed with warning

Migration has completed with some warnings. For more information, see the migration report.

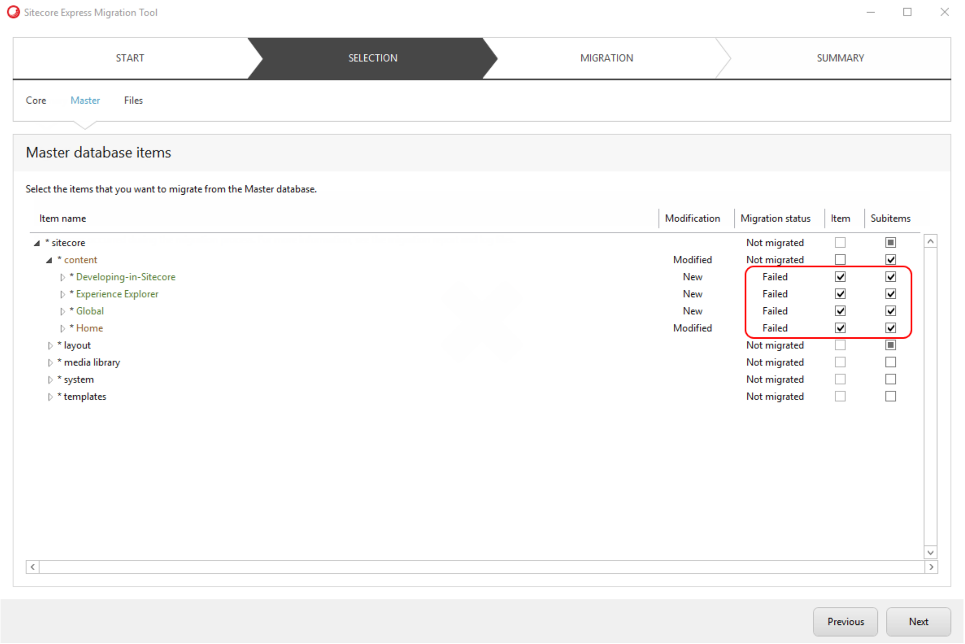
You must perform some post-migration steps and configure your Sitecore instance. For more information, see the Sitecore Express Migration Guide.

A Sitecore migration completed with a warning means that some items or files could not be migrated. If your migration is completed with a warning, you must open The migration report for more information.

Note
It is possible that although the migration finished with a warning, your new Sitecore installation can still work. To migrate the files that failed, you must re-run the Express Migration Tool.

The tool remembers and saves its state in the local database. If an error occurs or the migration process finished with an error, the tool retries to migrate the items that failed. This helps to reduce the number of errors that can occur during the migration process.

When you retry the migration, the tool skips any item or file that was already copied during the previous migration attempt, and in the file selection step, you can select or unselect the files or items that were not migrated.



Master database items

Select the items that you want to migrate from the Master database.

Item name	Modification	Migration status	Item	Subitems
◢ * sitecore		Not migrated	□	▣
◢ * content	Modified	Not migrated	□	☑
▷ * Developing-in-Sitecore	New	Failed	☑	☑
▷ * Experience Explorer	New	Failed	☑	☑
▷ * Global	New	Failed	☑	☑
▷ * Home	Modified	Failed	☑	☑
▷ * layout		Not migrated	□	▣
▷ * media library		Not migrated	□	□
▷ * system		Not migrated	□	□
▷ * templates		Not migrated	□	□

3.4 Known issues

3.4.1 Configuration migration

Known issue

Although the Express Migration Tool enables you to specify a custom location for the media folder, there is one exception to this rule.

If you store the media folder on the same server as the source instance, for example, `D:\MediaFiles`, and the `Media.FileFolder` setting is set to the local path `D:\MediaFiles`, when you run the tool from another server, it searches for the media files in `D:\MediaFiles` on that server, which is not correct.

To avoid this, you must give Server Message Block (SMB) access to the media folder on your network, and in the `web.config` file, set the `Media.FileFolder` setting to the relevant network path before running the tool.

Known issue

The Express migration tool ignores the `<assemblyBinding>` node in the `web.config` file. This means that the tool does not generate `diff` files for any changes made to this node or its descendants.

```
        <configuration>
          <runtime>
            <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="0.0.0.0-3.5.0.0" newVersion="4.5.0.0" />
              </dependentAssembly>
              <dependentAssembly>
                <assemblyIdentity name="System.Web.Mvc" publicKeyToken="31bf3856ad364e35" 
xmlns="urn:schemas-microsoft-com:asm.v1" />
                <bindingRedirect oldVersion="1.0.0.0-5.0.0.0" newVersion="5.1.0.0" 
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>
            </assemblyBinding>
          </runtime>
        </configuration>
```

Known issue

When a change is made in the `Sitecore` section of the `web.config` file, the Express Migration Tool generates the following patch file: `sitecore.section.patch.config.disabled` located in the `<target instance root folder>\Migration.MigratedConfiguration\App_Config` folder.

However, because of the design of the patching mechanism, before applying the patch, you must copy the file from the `\App_Config` folder to the `\App_Config\Include` folder.

Known issue

Configuration nodes that do not have a unique ancestor are patched incorrectly. Therefore, if there is a configuration node that does not have a unique ancestor on the source instance, and there is a change in the following configuration line:

```
        <configuration>
          <sitecore>
            <someNodeContainer>
              <child type=”Sitecore.ClassName, SitecoreAssembly”>
```

```
                <key>first</key>
                <param ref=”component1” />
              </child>
              <child type=”Sitecore.ClassName, SitecoreAssembly”>
                <key>second</key>
                <param ref=”overridenComponent” />   //THIS IS THE MODIFIED LINE
              </child>
              <child type=”Sitecore.ClassName, SitecoreAssembly”>
                <key>third</key>
                <param ref=”component3” />
              </child>
            </someNodeContainer>
          </sitecore>
        </configuration>
```

the tool generates the following patch:

```
        <configuration xmlns:patch=”…”>
          <sitecore>
            <someNodeContainer>
              <child type=”Sitecore.ClassName, SitecoreAssembly”>
                <param ref=” component2”>
                  <patch:attribute name=”ref” value=”overridenComponent” />
                </param>
              </child>
            </someNodeContainer>
          </sitecore>
        </configuration>
```

When this happens, the first child node is patched instead of the second child node.

Known issue

Generating `diff` files is not supported for configuration files that contain non-default namespaces. In Sitecore 7.2 there are two files that contain those nodes:

- Domains.config
- GlobalRoles.config

The tool ignores any changes that have been made to the `<sc:templates>` node in both of the `config` files.

Known issue

After you install a module on the comparison instance, the Express Migration Tool can identify the module items as *Modified*. This is because during the installation, the parent ID can be changed and the tool then identifies the module as being *Added* or *Deleted*, and the children items are identified as *Modified*.

In other cases, the Express Migration Tool can identify some items as *Added* and *Deleted* because they are not included in the module package and are automatically created by the Sitecore installer when you install the module.

3.4.2 Security roles migration

Known issue

If a Sitecore administrator deleted a specific security role on the source instance before migration, the role is not deleted on the target instance after migration is completed. This is done in order to protect the target’s system code from breaking changes.

After migration is completed, you can manually delete this role if you want to, but you must conduct all relevant tests to ensure that there are no breaking changes.

Chapter 4

Post-migration steps

This chapter describes the necessary steps to complete the migration.

This chapter contains the following sections:

- Post-migration steps
- Migration FAQs

4.1 Post-migration steps

If you want the target instance to function in the same way as the source instance, you must perform post-migration steps. For example, you must configure the instance and recompile your code against the new version of the platform.

You must complete the following post-migration steps:

- Configure the target instance.
- Rebuild your code and manage any breaking changes in custom functionality.
- Restart the application pool in IIS for the target instance.
- Rebuild the Search Indexes and the Link Database.
- Clean up the Core and Master databases.

The following post-migration step is optional:

- Run the broken links report and fix any broken links.

4.1.1 Configure the target instance

The way you configure the target instance can vary from instance to instance.

The tool generates patch and `diff` files to help you configure the target instance. However, the configuration of the target instance is not completely automated and you must review the configuration patches, to ensure they are valid for the target instance, and apply them as necessary.

To configure the target instance:

1. In the `<target instance root folder>\Migration.MigratedConfiguration` folder, find the configuration patch and `diff` files.

   The file name and location represent the name and location of the relevant file on the source instance.

   For example:

   - For the source configuration file, located in the `\App_Config\Include\File_Name.config` folder, the tool generates a patch file with the `.disabled` extension on the target instance, in the `<target instance root folder>\Migration.MigratedConfiguration \App_Config\Include\File_Name.patch.config.disabled` folder
   - If there are changes to the `web.config` file, the tool either creates the patch file for the Sitecore section in the, `\Migration.MigratedConfiguration\App_Config\sitecore.section.patch .config.disabled` folder, or the `diff` file for non-Sitecore sections, for example `<target instance root folder>\Migration.MigratedConfiguration\web.diff.config.disabled`

2. In a text editor, open the `diff` files generated by the tool, find the commented nodes and copy the changes to the appropriate files on the target instance.

   The XML comments identify breaking changes. The comments identify and explain the changes that were made to the specific node on your source instance compared to the target instance, and identify if the node was modified by Sitecore in subsequent releases.

3. In a text editor, open the patch files and look for the configurations that have breaking changes. If necessary, edit the patch file to accommodate the changes.

4. Copy the patch files to the `\App_Config\Include` folder, and enable them by removing the `.disabled` extension from the file name.

   Note
   You can find the configuration file changes document for all versions of Sitecore on the Sitecore Developer Portal and the Sitecore Developer Network.

Editing a patch file

To edit a patch file:

1. In a text editor, open each of the migrated config files and find the XML comments added by the tool that have a hyperlink to the Sitecore Developer Portal – https://dev.sitecore.net.

   For example:

```
        <!--EXPRESS MIGRATION: The node has been added-->
        <!--This configuration node was modified or deleted in your source instance and
it has also been modified or deleted in a subsequent Sitecore release.-->
        <!--For more information, see the configuration file changes document:
https://dev.sitecore.net/Downloads/Sitecore_Experience_Platform/Sitecore_81/Sitecore_E
xperience_Platform_81_Update1.aspx
```

   The XML comments in the migrated config files are very important because they show the config file changes between the different versions of Sitecore.

2. In your browser, enter the URL that you found in the XML comments, open the config file changes document, and find the relevant configuration next to the comments.

   For example: `x:attribute name=”addAspxExtension”>false</x:attribute>`

   In the XML comments, in the relevant nodes, depending on the changes made by Sitecore between versions, you must either modify, delete, or do nothing to the file.

4.1.2 Rebuilding custom functionality

The Express Migration Tool identifies and copies the new and modified files from the source instance to the file system in the target instance, including files related to layouts, renderings, and program files.

However, any custom code or customizations that existed before the migration process may not work correctly after migration is completed. This is because of changes made to the platform and API in Sitecore since version 7.2.

When migration is completed, you must recompile your code and manage any breaking changes that occur. You can find a list of all the changes that were made in the release documentation for each Sitecore version on the Sitecore Developer Network and on the Sitecore Developer Portal.

4.1.3 Restart the application pool

To complete the migration, you must reset all Sitecore caches so that changes made to the databases take effect.

To do this, in the Internet Information Services Manager (IIS) you must stop and restart the application pool for the appropriate Sitecore target instance.

4.1.4 Clean up the databases

To clean up the databases:

1. In the Sitecore Launchpad, click **Control Panel**, and in the **Database** section, click **Clean up databases**.
2. In the **Clean up Databases** dialog box, select the Core and Master databases and click **Clean** to run the cleanup process.

4.1.5 Rebuild the link databases and run the Broken Links report

The Broken Links report scans the database and locates broken Sitecore links. If the system finds any broken links, you can update and rebuild the link databases for the Master and Core databases.

To rebuild the link databases:

- Log in to the Sitecore Desktop and click **Control Panel**, **Database**, **Rebuild the Link Database**, and select all the available databases.

To run the Broken Links report:

1. In the Sitecore Launchpad, click **Control Panel**, and in the **Reports** section, click **Scan the database for broken links**.
2. In the **Scan for Broken Links** dialog box, in the **Databases** group, select the **core**, **master**,or **web** databases check box.

   Note
   If you want to scan for broken links in your system files, in the **Scan for Broken Links** dialog box, in the **View** group, select the **System Items** check box.

3. In the **Home** group, click **Scan**.

4.1.6 Rebuild search indexes

To complete the migration process, you must rebuild your search indexes and the Link database.

To rebuild all the ContentSearch indexes:

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

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

1. In the Sitecore Launchpad, click **Control Panel**, and in the **Database** section, click **Rebuild search indexes**.
2. Select the **Quick search index** check box, and then click **Rebuild**.

4.2 Migration FAQs

Q: Does the tool keep logs of the changes that it makes?

A: Yes, you can find the log files in the `\ Sitecore Express Upgrade Tool\Logs` folder.

Q: Can I run the same instance of the tool to upgrade two different servers?

A: Yes.

However, this is not recommended practice. The tool was not designed to make use of the same instance of the tool to upgrade different sources.

You should always use a clean instance of the tool to migrate another source instance.

The Sitecore Express Migration Tool remembers and saves its state in the local database as well as the states of all objects after the previous migration was completed.

You can see the states of all objects on the **Selection** progressive tab. As a result, if you run the tool for a second or subsequent time, and you have made changes to items and/or files in the source instance, and you want to migrate these changes, you must re-run the analysis.

This is because the tool collects all the information about the differences between the source and comparison instances in the Analysis step.

When you restart the analysis, the tool deletes all the information about the previous migration attempt and tries to migrate items and files that have already been migrated, unless you unselect them in the Selection step.

Q: Does the tool support Sitecore Analytics to xDB migration?

A: No.

The first version of the Sitecore Express Migration Tool does not support Sitecore Analytics to xDB migration. However, you can either start the migration process over again, start your analytics and xDB from the blank sheet on the target instance, or you can use the Analytics Conversion Tool provided with the Sitecore Experience Platform. You can download the tool from the Sitecore Developer Network.

When you use the Analytics Conversion Tool, you must gradually convert the analytics databases from Sitecore 7.2 to 7.5, to 8.0, to 8.1, and finally to Sitecore 8.2.

Q: Does the tool automatically migrate Sitecore modules?

A: No.

The first version of the tool does not support automatic migration of modules. However, the tool identifies all the new items and files on the source instance, and you can select the content items of the module and migrate them to the target instance.

Note
During migration, the tool does not perform an automatic upgrade of the modules' content items.

Q: What if the migration process fails?

A: If the migration process fails, you have two options:

- Retry migration.

  The tool saves its state in the local database. If the previous procedure stopped with an error, you can run the tool again. At this point, the tool tries to migrate the items that were not migrated in the previous attempt.

Note
Before you restart the migration process, you must clean up the tool's local databases or run a new instance of the tool.

To clean up the tool's local databases, delete the `{Express Migration Root folder}/Migration.db` file.

- Restart the migration process.

  The tool does not make changes to the source instance but it may have copied some files to the target instance. You should therefore create a new target instance and start the process again.