Express Migration Guide 3.0

Sitecore XP 9.0

How to upgrade from Sitecore 6.6, 7.2, 7.5 and 8.0 directly to Sitecore 9.0

sitecore®
Own the experience™

Table of Contents

Chapter 1

Introduction

This document describes how to use and execute the Express Migration Tool 3.0 rev. 171011.

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 any errors.
- **Chapter 4 – Post-migration Steps**
  Step-by-step instructions for completing the migration process.

1.1 Getting Started

The Sitecore Express Migration Tool 3.0 is designed to help you migrate directly from older versions of Sitecore to the latest version of Sitecore Experience Platform™ 9.0 rev. 171002 (Initial Release). The Sitecore instance that you are migrating to is called the target instance.

The Express Migration Tool 3.0:

- 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.
- Lets you migrate custom modules and the Web Forms for Marketers module.

1.1.1 Supported Versions

The Sitecore Express Migration Tool 3.0 enables you to migrate from older Sitecore versions to the latest version of the Sitecore Experience Platform™.

You can upgrade from the following versions of Sitecore:

- Sitecore 8.0 rev. 160115 (8.0 Update-7)
- Sitecore 8.0 rev. 151127 (8.0 Update-6)
- Sitecore 8.0 rev. 150812 (8.0 Update-5)
- Sitecore 8.0 rev. 150621 (8.0 Update-4).
- Sitecore 8.0 rev. 150427 (8.0 Update-3).
- Sitecore 8.0 rev. 150223 (8.0 Update-2).
- Sitecore 8.0 rev. 150121 (8.0 Update-1).
- Sitecore 8.0 rev. 141212 (8.0 Initial Release).
- Sitecore 7.5 rev.150212 (Update-2)
- Sitecore 7.5 rev.150130 (Update-1)
- Sitecore 7.5 rev.141003 (Initial Release)
- 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)
- Sitecore 6.6.0 rev. 140410 (6.6.0 Service Pack-2, originally released as 6.6 Update-8
- Sitecore 6.6.0 rev. 131211 (6.6.0 Update-7)
- Sitecore 6.6.0 rev. 130529 (6.6.0 Service Pack-1 Update-6)
- Sitecore 6.6.0 rev. 130404 (6.6.0 Update-5)
- Sitecore 6.6.0 rev. 130214 (6.6.0 Update-4)
- Sitecore 6.6.0 rev. 130111 (6.6.0 Update-3)

- Sitecore 6.6.0 rev. 121203 (6.6.0 Update-2)
- Sitecore 6.6.0 rev. 121015 (6.6.0 Update-1)
- Sitecore 6.6.0 rev. 120918 (6.6.0 Initial Release)

To the following version of Sitecore:

- Sitecore Experience Platform™ 9.0 rev. 171002 (Initial Release).

Note
The Express Migration Tool 3.0 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.

1.1.2 Supported Modules

The Express Migration Tool 3.0 enables you to migrate custom modules and the Web Forms for Marketers (WFFM) module.

You can migrate from the following versions of Web Forms for Marketers:

- Web Forms for Marketers 2.4 rev. 140117 running on Sitecore 7.2
- Web Forms for Marketers 2.4 rev. 140923 running on Sitecore 7.2
- Web Forms for Marketers 2.4 rev. 141008 running on Sitecore 7.2
- Web Forms for Marketers 2.4 rev. 150619 running on Sitecore 7.2
- Web Forms for Marketers 2.4 rev. 151103 running on Sitecore 7.2

To the following version of the Web Forms for Marketers module:

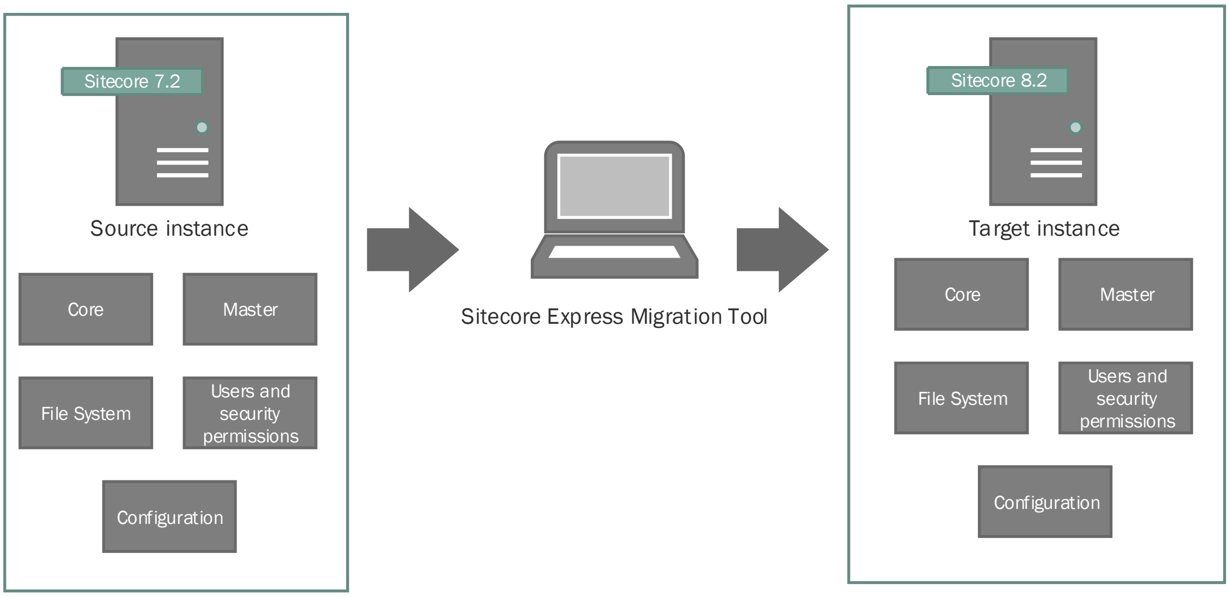
- Web Forms for Marketers 9.0 rev. 171003 running on Sitecore Experience Platform™ 9.0 rev. 171002.

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.

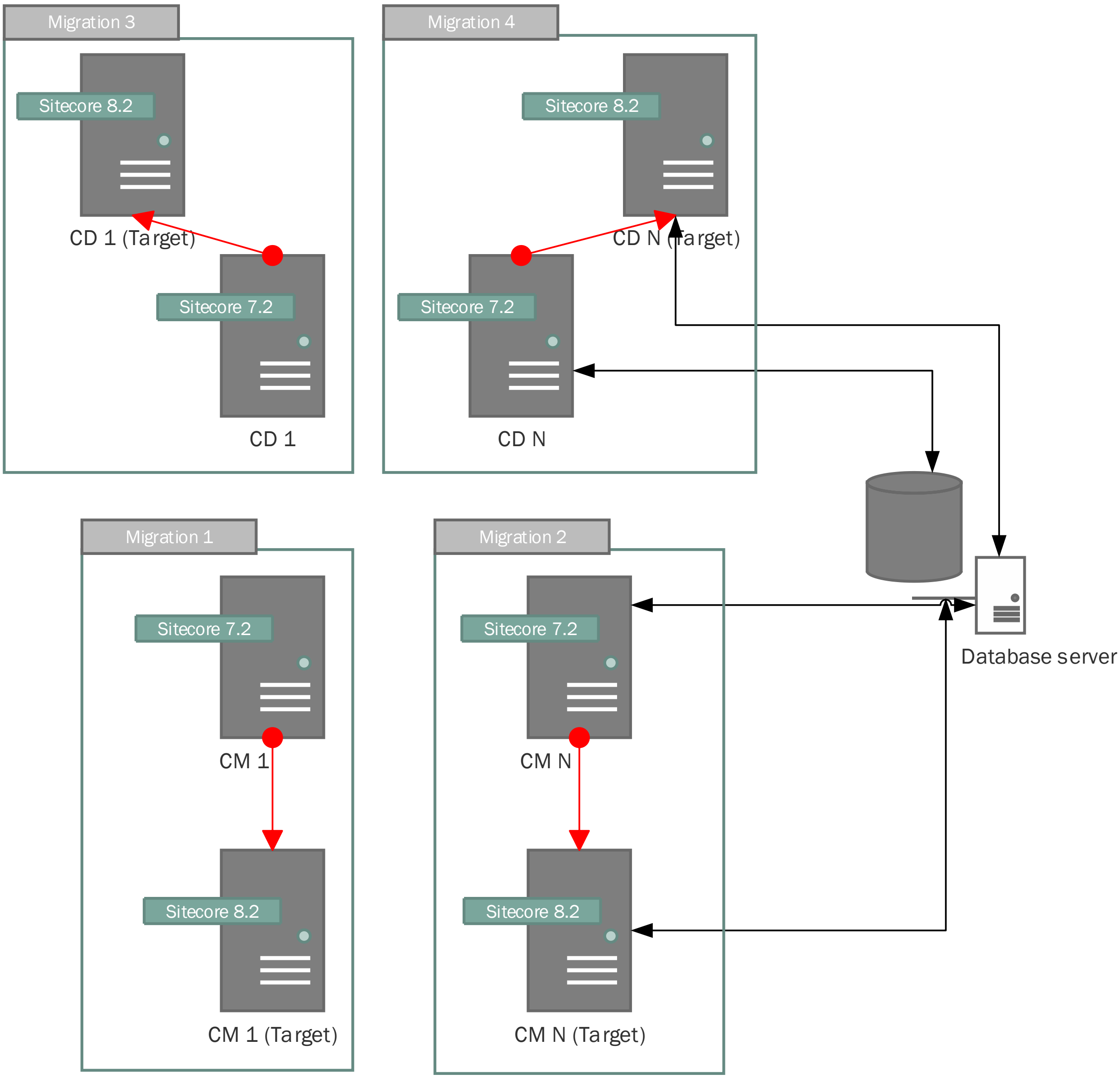


1.2.1 A Distributed Environment

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

One instance of the Sitecore Express Migration Tool can migrate one instance of Sitecore at a time. Therefore, in a scaled environment you can run the tool once and configure the remaining servers. The specific settings can vary from server to server.

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



Migration example

Use the Sitecore Express Migration Tool to:

- Migrate the content management server (CM 1) and configure the remaining content management servers
- Migrate the configuration and file system of these content management servers (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.

For the content delivery servers:

- Set up and configure the new content delivery servers. Use the Express Migration Tool 3.0 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 3.0.
- 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™ 9.0 rev. 171002 from the Sitecore Downloads page – 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 3.0 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 2016 (64-bit)
- Windows Server 2008 (64-bit)
- Windows Server 2008 R2 (64-bit)
- Windows Server 2012 (64-bit)
- Windows 2012 R2 (64-bit)
- Windows 10 (64-bit)
- Windows 8.1 (64-bit)
- Windows 8 (64-bit)
- Windows 7 (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.

  In addition, 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 let you configure the maximum number of retries and the 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).

The Express Migration Tool also lets you increase or decrease the number of threads for the execution of a parallel task depending on your hardware configuration (the default value is 100). You can also configure the maximum allowed concurrency level for the parallel migration of big blobs stored in a database (the default value is 3 threads). The tool identifies a blob item as big when it is more than 1 MB in size.

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 **Selection** tab.

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

1. In a text editor, edit the `Website\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 maximum allowed concurrency level for the execution of parallel tasks and the migration of big blobs stored in a database:

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

      `<setting name="MaxConcurrencyLevel" value="100" />`
    - Big blob maximum concurrency level:

      `<setting name="BigBlobMaxConcurrencyLevel" value="3" />`
3. Save your changes.

To configure the files and folders that you want the tool to ignore:

1. In a text editor, edit the `Website\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 you want the tool to 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 Downloads page – https://dev.sitecore.net.

To run the Express Migration Tool:

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
- Module Migration
- Known Issues

3.1 Using the Express Migration Tool

Before you run the Express Migration Tool 3.0, 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.

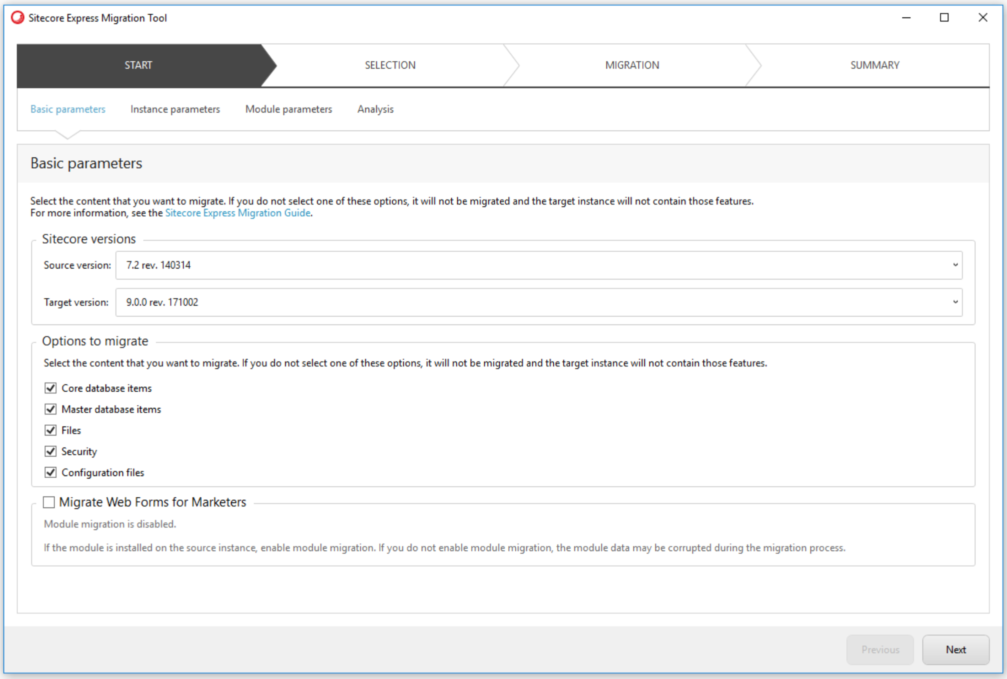
3.1.1 Starting the Migration Process

The Express Migration Tool contains four progressive tabs:

- Start – select the parts of the instance that you want to migrate, provide parameters of your source and target instances, and specify module-related options.
- Selection – select items and files in the Core and Master databases that you can migrate.
- Migration – monitor the progress as the tool migrates the data to the target instance.
- Summary – see information about whether the migration was succesful and access the migration report.

When you have installed and configured the tool, to start 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:



Basic 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.

Sitecore versions

Source version: 7.2 rev. 140314

Target version: 9.0.0 rev. 171002

Options to migrate

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.

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

□ Migrate Web Forms for Marketers

Module migration is disabled.

If the module is installed on the source instance, enable module migration. If you do not enable module migration, the module data may be corrupted during the migration process.

2. On the **Start** progressive tab, on the **Basic parameters** subtab, in the **Sitecore versions** section, in the **Source version** field, click the drop-down arrow and select the version of Sitecore you are migrating from.
3. In the **Target version** field, click the drop-down arrow and select the version of Sitecore you are migrating to.
4. In the **Options to migrate** section, select or clear the check boxes specifying the content from the source instance that you want to migrate.

   By default, all check boxes are selected.

   Note
   To enable module migration, in the **Migrate Web Forms for Marketers** section, you must select the **Migrate Web Forms for Marketers** check box. For more details about module migration, see the *Module Migration* chapter.

5. Click **Next**.

Specify Instance Parameters

When you specify instance parameters, you specify the connection strings for the Core and Master databases for the target and source instances.

The connection strings are stored in the `<YourInstanceFolder>\App_Config\ConnectionStrings.config` file of your Sitecore source or target instance. 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. Depending on how you have set up your instances, you can modify the connection strings after you enter them in the tool.

For example:

```
Data Source=(local);Initial Catalog=sc90rev171002Sitecore_core;Integrated Security=False;User
ID=sa;Password=Password1234
```

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

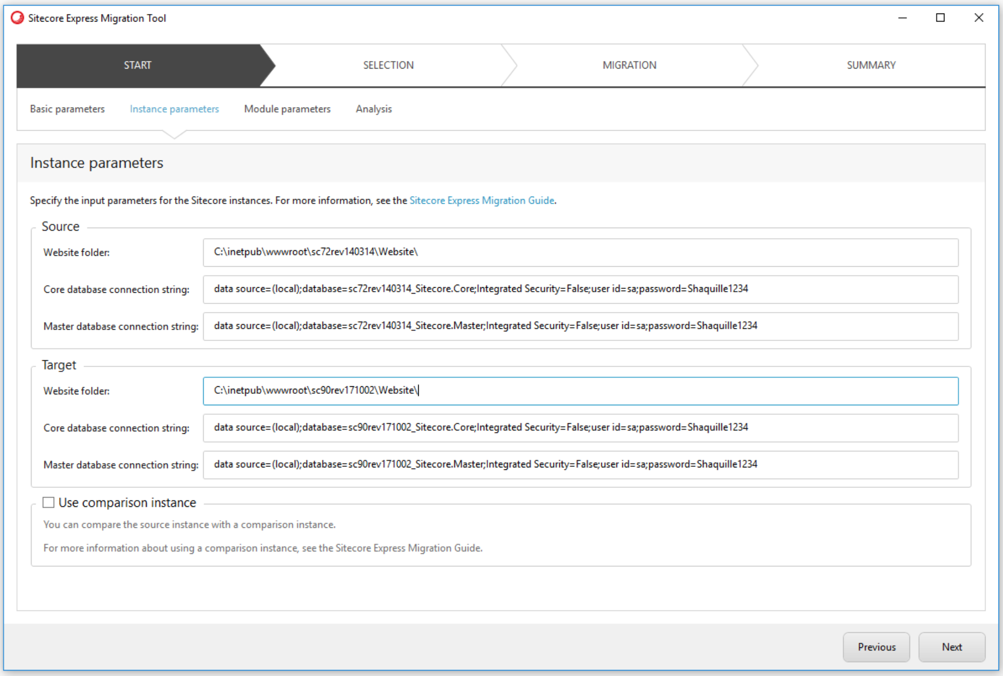
You can specify a non-local connection string for both the **Core database connection string** field, or the **Master database connection string** field. To do this, in the `Data Source` parameter, enter the network name of the server.

For example:

```
Data Source=ProdServer;Initial Catalog=sc90rev171002Sitecore_core;Integrated
Security=False;User ID=sa;Password=Password1234
```

To specify the instance parameters:

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



Instance parameters

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

Source

Website folder: C:\inetpub\wwwroot\sc72rev140314\Website\

Core database connection string: data source=(local);database=sc72rev140314_Sitecore.Core;Integrated Security=False;user id=sa;password=Shaquille1234

Master database connection string: data source=(local);database=sc72rev140314_Sitecore.Master;Integrated Security=False;user id=sa;password=Shaquille1234

Target

Website folder: C:\inetpub\wwwroot\sc90rev171002\Website\

Core database connection string: data source=(local);database=sc90rev171002_Sitecore.Core;Integrated Security=False;user id=sa;password=Shaquille1234

Master database connection string: data source=(local);database=sc90rev171002_Sitecore.Master;Integrated Security=False;user id=sa;password=Shaquille1234

□ Use comparison instance

You can compare the source instance with a comparison instance.

For more information about using a comparison instance, see the Sitecore Express Migration Guide.

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

```
Data Source=ProdServer;Initial Catalog=sc80rev141212Sitecore_core;Integrated
Security=False;User ID=sa;Password=Password1234
```

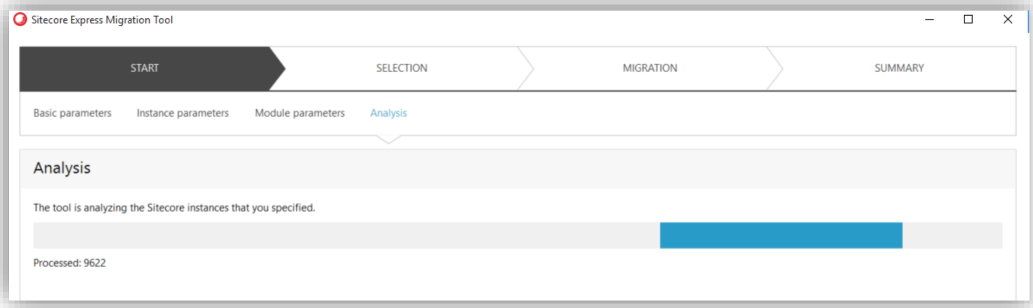
3. In the **Master database connection string** field, enter the connection string for the Master database on your source instance.
4. In the **Target** section, in the **Website folder** field, enter the path to your target instance.
5. In the **Core database connection string** field, enter the connection string for the Core database.
6. In the **Master database connection string** field, enter the connection string for the Master database.

Note
The Express Migration Tool also lets you use a comparison instance. For more details, see the chapter *Installing and Using a Comparison Instance* .

7. Click **Next**.

   If you selected to migrate modules, the Module parameters substep appears in the next window. For more details, see the *Module Migration* chapter.

   Alternatively, the tool begins the analysis of the source instance and identifies all the new and modified files, items, and so on.

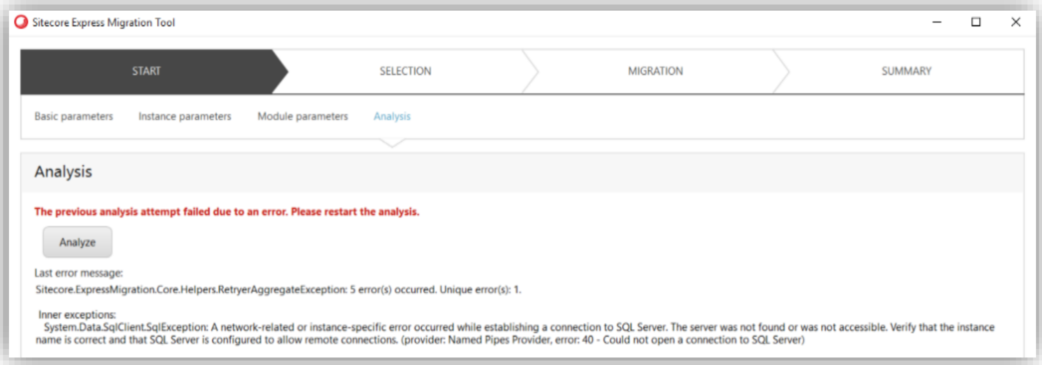


Analysis

The tool is analyzing the Sitecore instances that you specified.

Processed: 9622

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



Analysis

The previous analysis attempt failed due to an error. Please restart the analysis.

Last error message:
Sitecore.ExpressMigration.Core.Helpers.RetryerAggregateException: 5 error(s) occurred. Unique error(s): 1.

Inner exceptions:
System.Data.SqlClient.SqlException: A network-related or instance-specific error occurred while establishing a connection to SQL Server. The server was not found or was not accessible. Verify that the instance name is correct and that SQL Server is configured to allow remote connections. (provider: Named Pipes Provider, error: 40 - Could not open a connection to SQL Server)

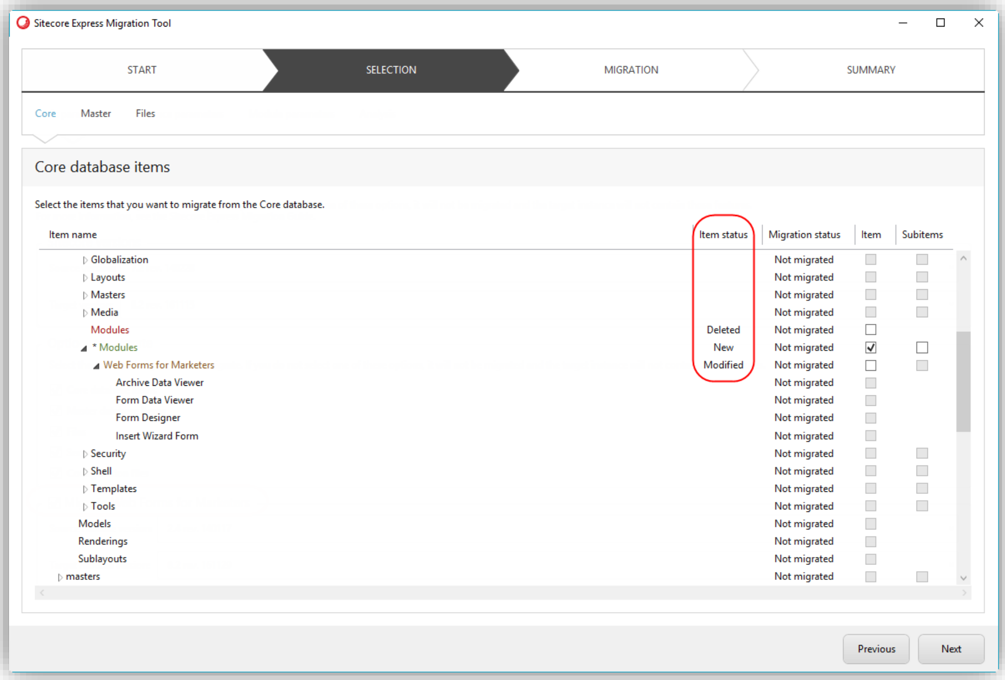
Selecting Items to Migrate

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



Core database items

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

Item name	Item status	Migration status	Item	Subitems
▷ Globalization		Not migrated	□	□
▷ Layouts		Not migrated	□	□
▷ Masters		Not migrated	□	□
▷ Media		Not migrated	□	□
Modules	Deleted	Not migrated	□	
◢ * Modules	New	Not migrated	☑	□
◢ Web Forms for Marketers	Modified	Not migrated	□	□
Archive Data Viewer		Not migrated	□	
Form Data Viewer		Not migrated	□	
Form Designer		Not migrated	□	
Insert Wizard Form		Not migrated	□	
▷ Security		Not migrated	□	□
▷ Shell		Not migrated	□	□
▷ Templates		Not migrated	□	□
▷ Tools		Not migrated	□	□
Models		Not migrated	□	
Renderings		Not migrated	□	
Sublayouts		Not migrated	□	
▷ masters		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:

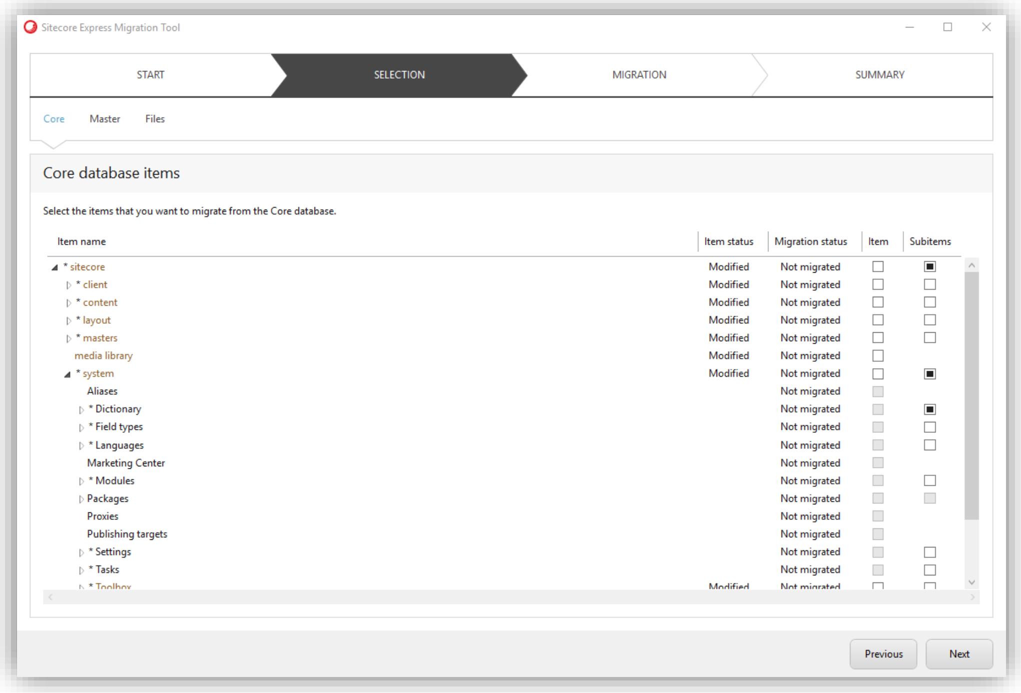
- New items are created in the target database.
- All of the files that are related to the selected media items are copied to the target instance.
- New files are copied to the target file system.
- Modified items are overwritten on the target instance.
- Deleted items and files in the source instance 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. On the **Selection** progressive tab, on the **Core** subtab, select the check box for items and subitems that you want to migrate, or clear the check box for items you do not want to migrate. Click **Next**.



Core database items

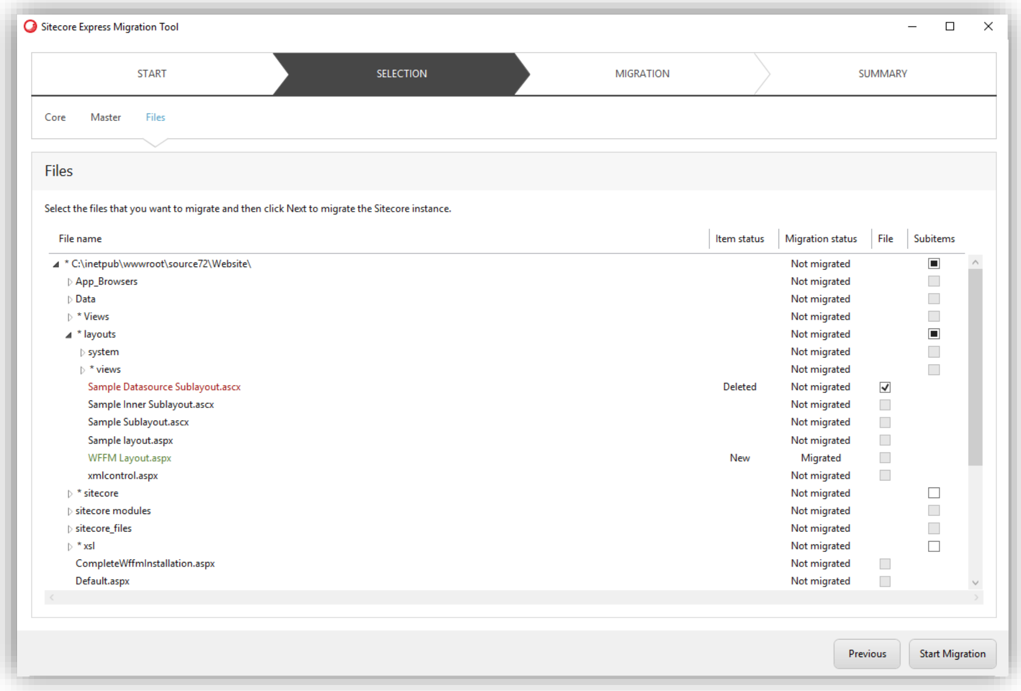
Select the items that you want to migrate from the Core database.

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
Items that contain modified and/or removed and/or new subitems are marked with an asterisk.

2. Repeat this procedure on the **Master database items** subtab and on the **Files** subtab.

   When you have finished selecting the items that you want to migrate, and to begin the migration process, click **Start Migration**.

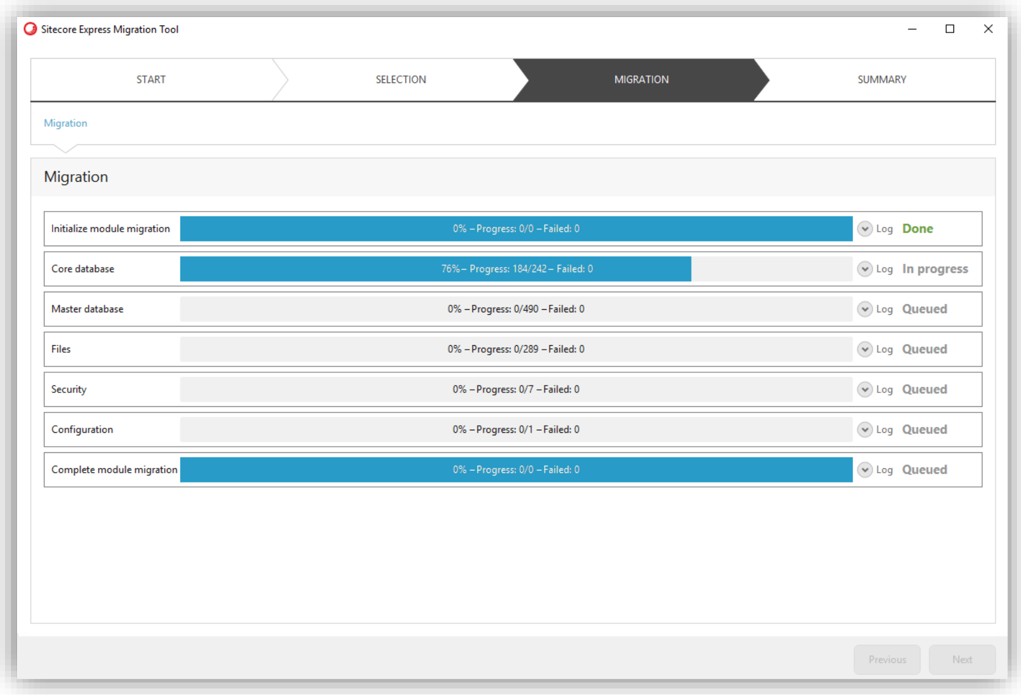


Files

Select the files that you want to migrate and then click Next to migrate the Sitecore instance.

File name	Item status	Migration status	File	Subitems
* C:\inetpub\wwwroot\source72\Website\		Not migrated		■
App_Browsers		Not migrated		□
Data		Not migrated		□
* Views		Not migrated		□
* layouts		Not migrated		■
system		Not migrated		□
* views		Not migrated		□
Sample Datasource Sublayout.ascx	Deleted	Not migrated	☑	
Sample Inner Sublayout.ascx		Not migrated	□	
Sample Sublayout.ascx		Not migrated	□	
Sample layout.aspx		Not migrated	□	
WFFM Layout.aspx	New	Migrated	□	
xmlcontrol.aspx		Not migrated	□	
* sitecore		Not migrated		□
sitecore modules		Not migrated		□
sitecore_files		Not migrated		□
* xsl		Not migrated		□
CompleteWffmInstallation.aspx		Not migrated	□	
Default.aspx		Not migrated	□	

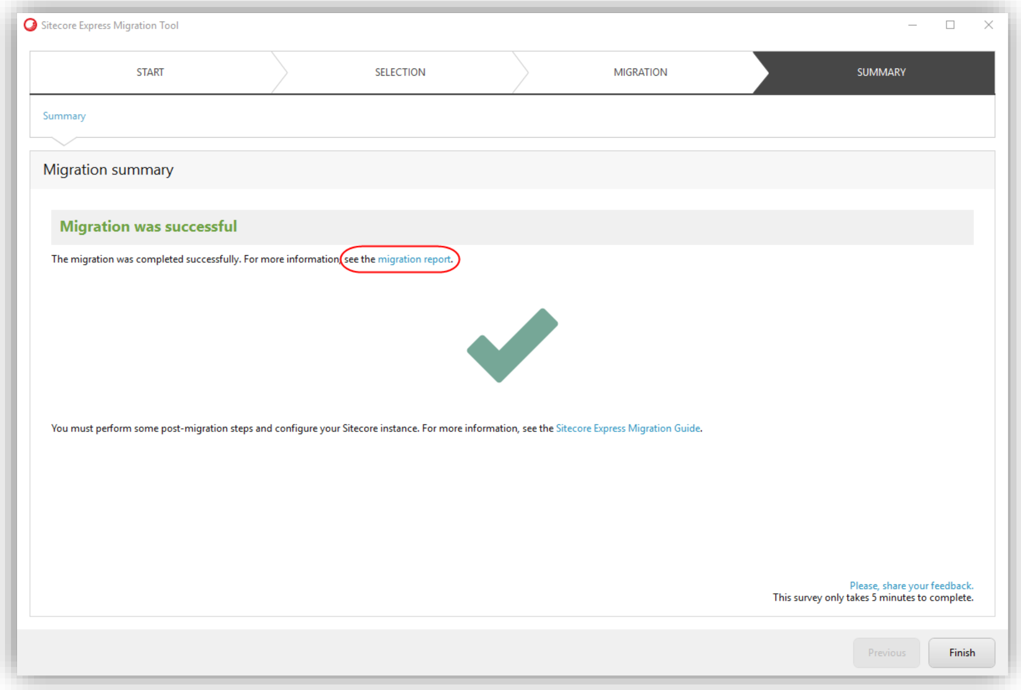
At this point, the Express Migration Tool begins to migrate the data to the target instance.



Migration

Initialize module migration	0% – Progress: 0/0 – Failed: 0	Log	Done
Core database	76% – Progress: 184/242 – Failed: 0	Log	In progress
Master database	0% – Progress: 0/490 – Failed: 0	Log	Queued
Files	0% – Progress: 0/289 – Failed: 0	Log	Queued
Security	0% – Progress: 0/7 – Failed: 0	Log	Queued
Configuration	0% – Progress: 0/1 – Failed: 0	Log	Queued
Complete module migration	0% – Progress: 0/0 – 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 **see 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.

Please, share your feedback.
This survey only takes 5 minutes to complete.

You have now successfully migrated your source instance to the target Sitecore 9.0 instance.

Tip
When the migration is complete, you can give feedback to the Sitecore development team about your migration experience. To do this, on the **Migration summary** page, click the **Please share your feedback** link. The feedback is anonymous, and the feedback form does not collect instance or license-specific information.

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.

If a migrated item with the same ID 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 or similar security groups as they were on the source instance.
- Some system roles from the source instance are not migrated, because they are irrelevant on the target version.
- Membership of some roles can be changed to make them relevant for the target instance.

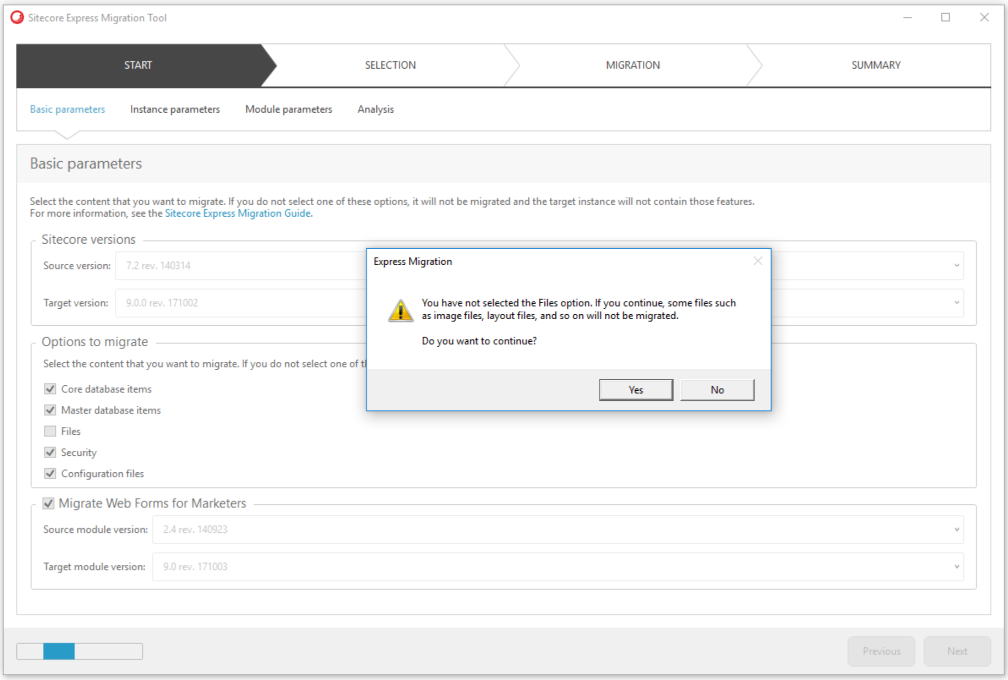
If a role with the same name exists on the source and on the target instances, the resulting role on the target instance has the same permissions and membership as the role on the source instance.

Files

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

Note
Any file(s) associated with your selected items (media files, rendering files, and so on) are also migrated to your target instance, even if you did not preselect them.

However, if you choose to migrate items and do not select the **Files** check box for migration, the tool displays a warning.



Basic 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.

Sitecore versions

Source version: 7.2 rev. 140314

Target version: 9.0.0 rev. 171002

Options to migrate

Select the content that you want to migrate. If you do not select one of t

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

☑ Migrate Web Forms for Marketers

Source module version: 2.4 rev. 140923

Target module version: 9.0 rev. 171003

Express Migration

You have not selected the Files option. If you continue, some files such as image files, layout files, and so on will not be migrated.

Do you want to continue?

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 `Website\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.

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 and files 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.

- To open the log files, navigate to the \Sitecore Express Upgrade Tool\Logs folder, and open the log*timestamp.txt file.

3.3 Installing and Using a Comparison Instance

The Express Migration Tool contains metadata about every revision of the supported source versions. When you run the tool, you must select the Sitecore version that you are currently using, or your source instance.

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 instance with a custom instance. This instance is called a *comparison instance*.

You can use a comparison instance in the following scenarios:

- 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.

  Note
  If you want to migrate the Web Forms for Marketers module, or other custom modules, you must use a comparison instance.

You can also use a comparison instance when:

- 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:

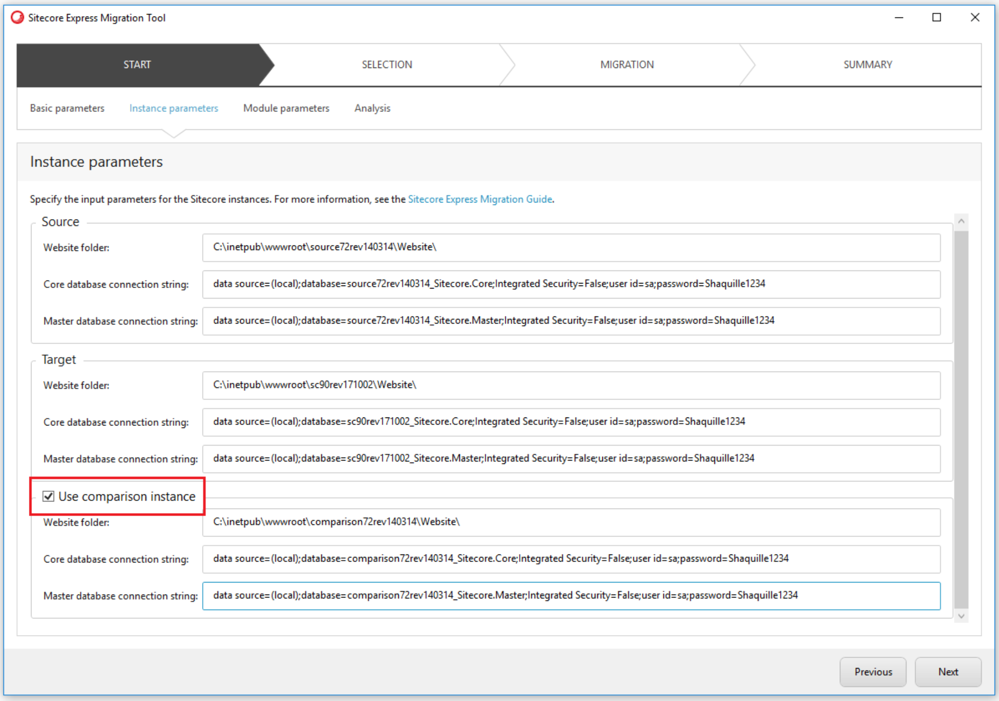
1. Install and configure the comparison Sitecore instance, and install all the required modules, patches, and files.

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

   Note
   You do not have to deploy the comparison instance.

2. To enable the use of a comparison instance, launch the Express Migration Tool, and on the **Start** progressive tab, on the **Instance parameters** subtab, select the **Use comparison instance** check box.

   For more details, see the *Specify Instance Parameters* section.



Instance parameters

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

Source

Website folder: C:\inetpub\wwwroot\source72rev140314\Website\

Core database connection string: data source=(local);database=source72rev140314_Sitecore.Core;Integrated Security=False;user id=sa;password=Shaquille1234

Master database connection string: data source=(local);database=source72rev140314_Sitecore.Master;Integrated Security=False;user id=sa;password=Shaquille1234

Target

Website folder: C:\inetpub\wwwroot\sc90rev171002\Website\

Core database connection string: data source=(local);database=sc90rev171002_Sitecore.Core;Integrated Security=False;user id=sa;password=Shaquille1234

Master database connection string: data source=(local);database=sc90rev171002_Sitecore.Master;Integrated Security=False;user id=sa;password=Shaquille1234

☑ Use comparison instance

Website folder: C:\inetpub\wwwroot\comparison72rev140314\Website\

Core database connection string: data source=(local);database=comparison72rev140314_Sitecore.Core;Integrated Security=False;user id=sa;password=Shaquille1234

Master database connection string: data source=(local);database=comparison72rev140314_Sitecore.Master;Integrated Security=False;user id=sa;password=Shaquille1234

3. In the **Use comparison instance** section, in the **Website folder** field, specify the path to your comparison instance.
4. In the **Core database connection string** field, enter the connection string for the Core database.

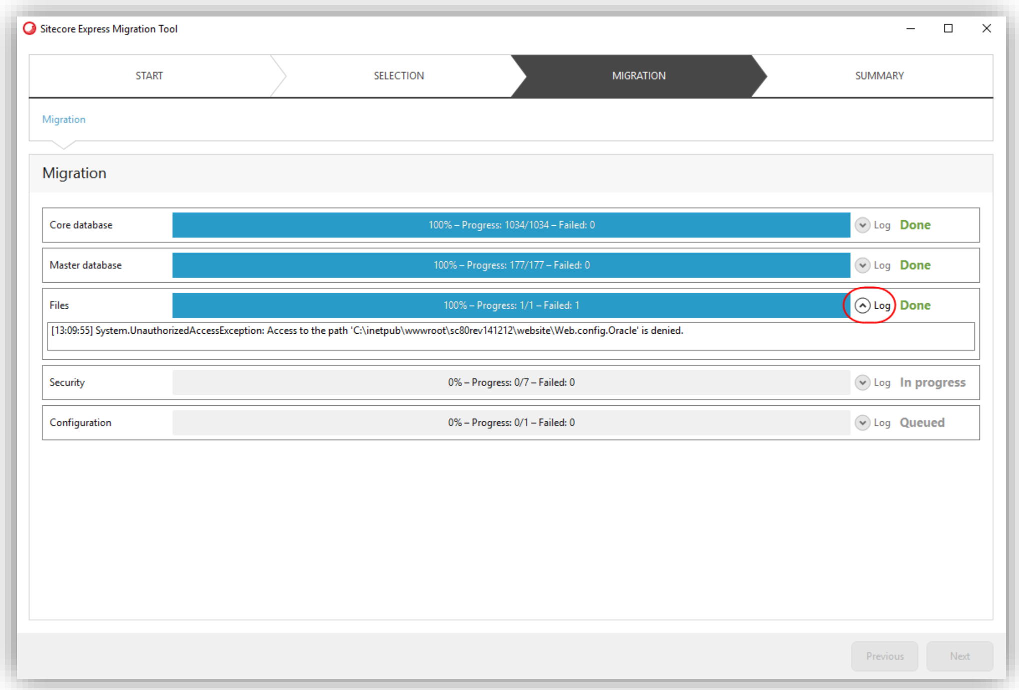
   The connection strings for the Core and Master databases are stored in the `\Website\App_Config\ConnectionStrings.config` file of your Sitecore comparison instance.
5. In the **Master database connection string** field, enter the connection string for the Master database.
6. Click **Next**.

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

3.4 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, on the **Migration** progressive tab, 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.



Migration

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

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

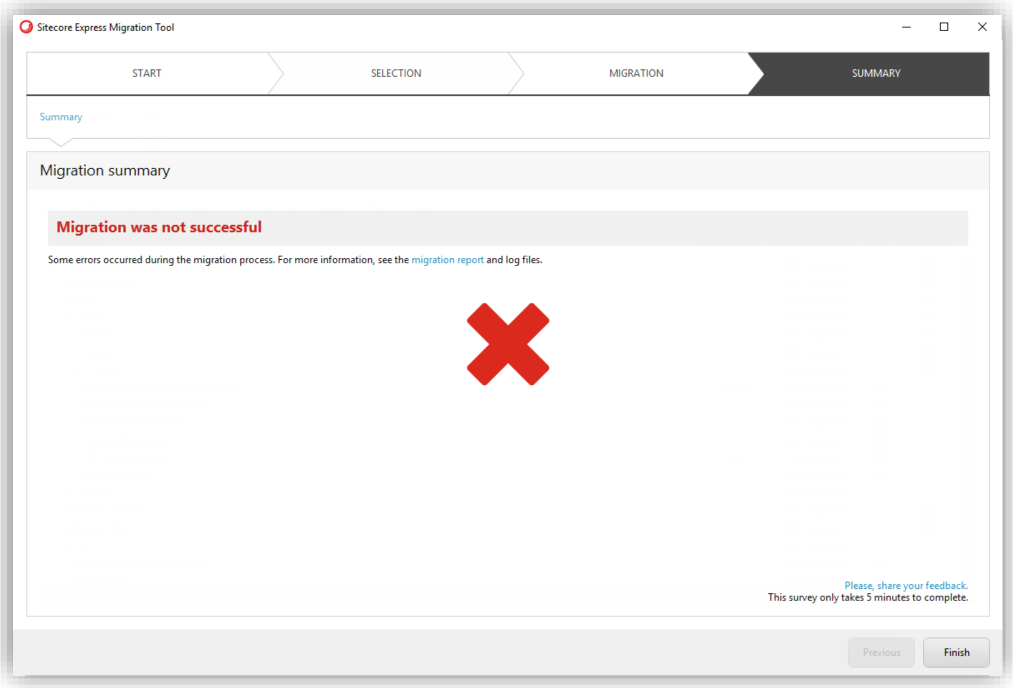
Files 100% – Progress: 1/1 – Failed: 1 Log Done

```
[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 and you want to read the migration report, click **migration report**. For more details about the report and what it contains, see *The Migration Report* section.



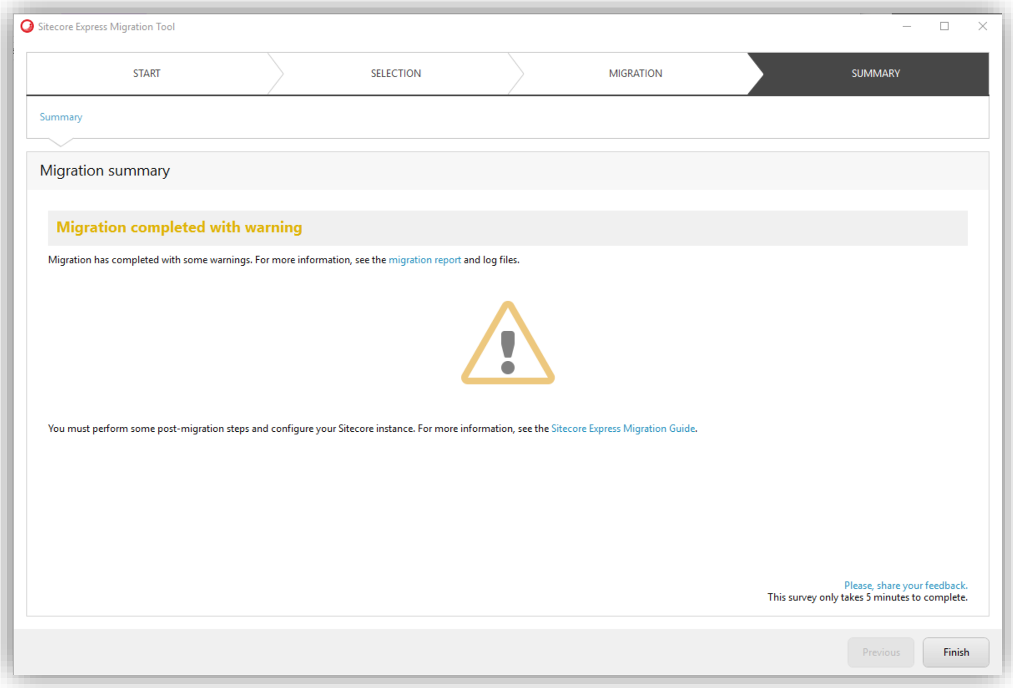
Migration summary

Migration was not successful

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

Please, share your feedback.
This survey only takes 5 minutes to complete.

The tool can also finish the migration process with warning(s).



Migration summary

Migration completed with warning

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

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

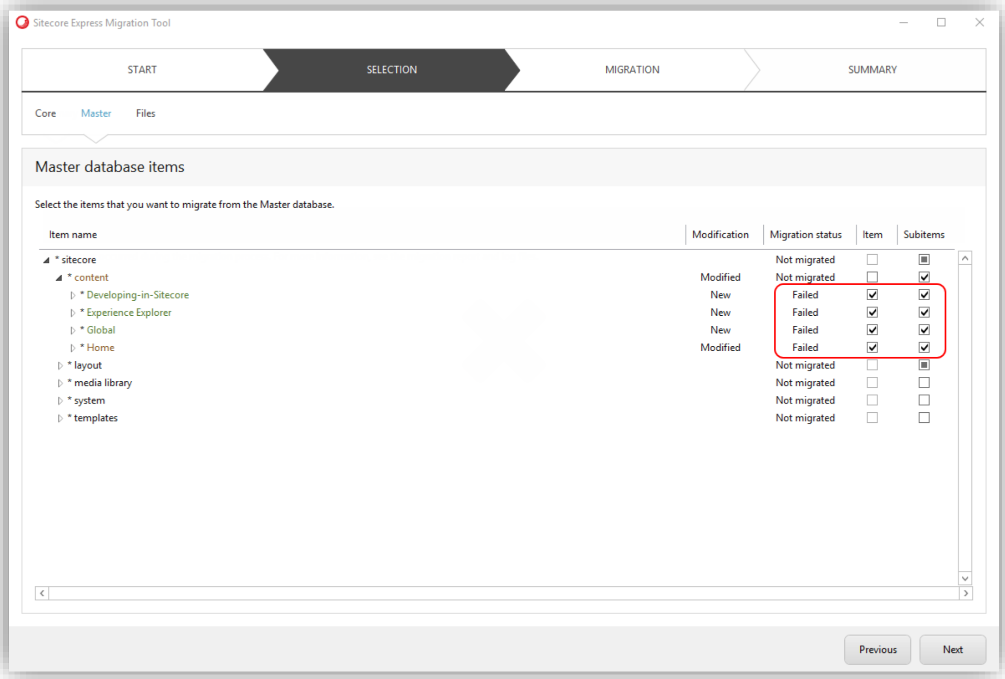
Please, share your feedback.
This survey only takes 5 minutes to complete.

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 may still work. To attempt to migrate the files, items, and configuration 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 on the **Selection** tab, on the **Core**, **Master**, and **Files** subtabs, you can select or clear 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	□	□

3.5 Module Migration

3.5.1 Web Forms for Marketers Migration

The Express Migration Tool 3.0 supports the migration of the Web Forms for Marketers module, in addition to the support for the migration of custom modules.

If you are migrating from any version of Sitecore 7.2, and you are running Web Forms for Marketers 2.4, the Express Migration Tool helps you migrate to the latest version of Web Forms for Marketers 9.0 rev. 171003, running on Sitecore Experience Platform™ 9.0 rev. 171002 (Initial Release).

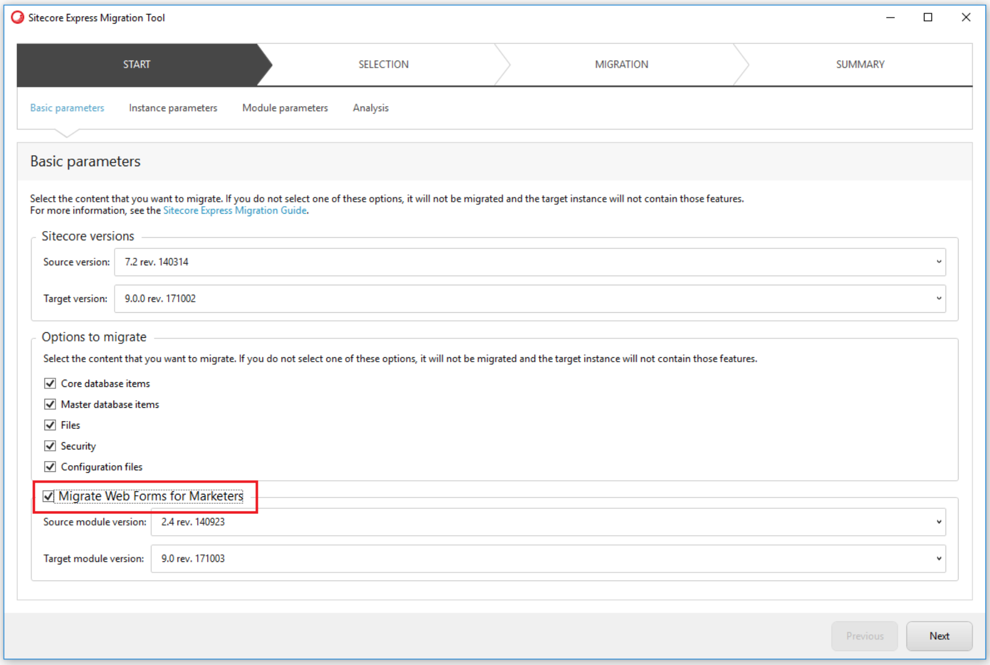
Note
This version of the tool only supports the migration of forms' data to a SQL Server-based Web Forms for Marketers setup.

To migrate the Web Forms for Marketers module:

1. Set up the comparison instance as described in the *Installing and Using a Comparison Instance* chapter.
2. Install the same version of the Web Forms for Marketers module on the comparison instance as the one you are already running on the source instance.
3. Install the latest version of the Web Forms for Marketers module on the target instance.
4. In the target instance, in the `\App_Config\Modules\WFFM\Sitecore.WFFM.Dependencies.config` file, set the Web Forms for Marketers data provider as SQL-server based:

   ```
   <formsDataProvider ref="/sitecore/wffm/analytics/sqlFormsDataProvider"/>
   ```

5. Run the Express Migration Tool and enable the migration of modules. To do this, on the **Start** progressive tab, on the **Basic parameters** subtab, select the **Migrate Web Forms for Marketers** check box.



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.

Sitecore versions

Source version: 7.2 rev. 140314

Target version: 9.0.0 rev. 171002

Options to migrate

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.

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

☑ Migrate Web Forms for Marketers

Source module version: 2.4 rev. 140923

Target module version: 9.0 rev. 171003

6. In the **Source module version** field, click the drop-down list and select the source version of the Web Forms for Marketers module.

7. In the **Target module version** field, click the drop-down list, select the target version and then click **Next**.

8. On the **Instance parameters** subtab, enter the connection strings and website folder paths for the source, target, and comparison instances as described in the *Specify Instance Parameters* section of this guide. Click **Next**.

Configuring Module Parameters

1. On the **Start** progressive tab, on the **Module parameters** subtab, in the **Source instance** section, in the **Database type** field, click the drop-down arrow, and depending on your Web Forms for Marketers database engine, select either *SQL Server* or *SQLite* source.

   Note
   For more details about configuring the SQLite database, and the pre- and post-migration steps, see the *Configuring and Migrating the SQLite WFFM* Database section.

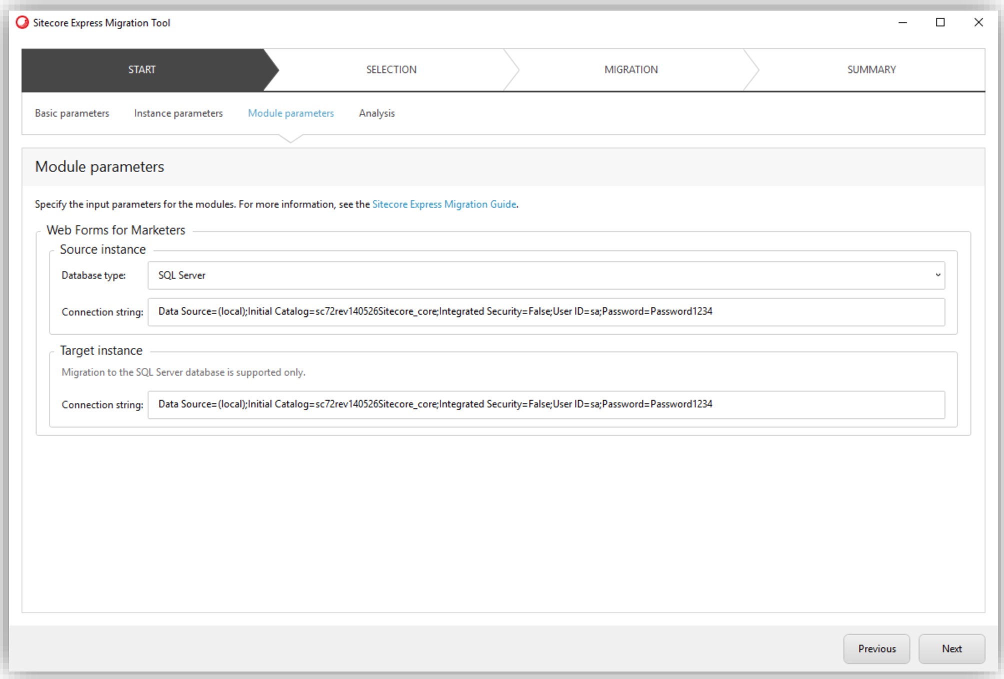
2. In the **Connection string** field, enter the source instance connection string to the Web Forms for Marketers database.

   You can find the connection string for the source instance in the `\Website\App_Config\Include\Sitecore.Forms.config` file of your source instance, in the following node:

   ```
   configuration/formsDataProvider/param[@desc= "connection string"] node inner text
   ```

3. In the **Target instance** section, in the **Connection string** field, enter the target instance connection string.

   You can find the connection string for the target instance in the `\App_Config\ConnectionStrings.config` file of your target instance.



Module parameters

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

Web Forms for Marketers

Source instance

Database type: SQL Server

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

Target instance

Migration to the SQL Server database is supported only.

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

4. Click **Next** and complete the migration as normal.

   Note
   The Web Forms for Marketers plugin handles the migration of the forms' content items and collected data. In the same way as any other configuration file, you can find the generated configuration

patches and diff files in the `<target instance root folder>\Migration.MigratedConfiguration` folder.

3.5.2 Configuring and Migrating the SQLite WFFM Database

If you have a SQLite database for your Web Forms for Marketers module that is larger than 1 GB in size, and / or you have more than 1.5 million records, you might want to configure the database to improve the speed of the migration process.

To do this, there are steps you must take before and after the migration process.

Before Migration

Before migration, in the target instance, you must disable the size limitation of the transaction log file, and replace clustered indexes in both tables in the Web Forms for Marketers module with non-clustered ones.

To limit the size of the transaction log file:

1. In SQL Server Management Studio, in the **Object Explorer** pane, expand the *Databases* folder.
2. Right-click the relevant database, and click **Properties**.
3. In the **Database Properties** window, in the **Select a page** pane, click **Files**.
4. In the **Autogrowth / Maxsize** column, set the maximum file size of the LDF file to *Unlimited*.

To adjust indexes:

1. In the `FormData` table, replace the `PK_FormData` clustered index with a similar, non-clustered index.
2. In the `FieldData` table, replace the `PK_FieldData` clustered index with a similar, non-clustered index.

   The following SQL Script replaces the indexes:

   ```
   CREATE UNIQUE NONCLUSTERED INDEX [NonClustered-Id] ON [FormData] ([Id] ASC)
   GO

   ALTER TABLE [FormData] DROP CONSTRAINT [PK_FormData]
   GO

   CREATE UNIQUE NONCLUSTERED INDEX [NonClustered-Id] ON [dbo].[FieldData] ([Id] ASC)
   GO

   ALTER TABLE [FieldData] DROP CONSTRAINT [PK_FieldData]
   GO
   ```

After Migration

After migration has completed, you can undo the changes you made to the indexes, and return them to their original state.

Note
Although the clustered indexes are the default configuration, the Web Forms for Marketers databases can also work with non-clustered indexes without affecting functionality. Therefore, it is optional to revert the indexes to their original state.

Depending on the size of your database, this operation can take a long time to complete.

The following SQL Script can undo the changes made to the indexes:

```
DROP INDEX [NonClustered-Id] ON [FormData]
GO

ALTER TABLE [FormData] ADD CONSTRAINT [PK_FormData] PRIMARY KEY CLUSTERED ([Id] ASC)
GO
```

```
DROP INDEX [NonClustered-Id] ON [FieldData]
GO

ALTER TABLE [FieldData] ADD CONSTRAINT [PK_FieldData] PRIMARY KEY CLUSTERED ([Id] ASC)
GO
```

3.5.3 Module Migration Plugins

The Express Migration Tool provides an API for third-party developers to extend the tool with plugins. You can use the plugins to create any custom module migration. For more information about plugin creation and an API reference, see the Express Migration Plugin Guide.

3.6 Known Issues

3.6.1 Items and Files 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.

3.6.2 Configuration Migration

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 `Website\App_Config` folder to the `Website\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

If you made any customizations to the Page Editor or the SheerUI ribbon of the Experience Editor, for example, adding buttons to the ribbon in the Experience Editor (formerly known as Page Editor in earlier versions of Sitecore), you must reimplement your customizations when you upgrade to Sitecore XP 9.0.

This is due to the SPEAK-based ribbon of the Experience Editor.

You can find the steps to implement the custom buttons in the Sitecore Documentation site – https://doc.sitecore.net

3.6.3 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.

3.6.4 Module Migration

Known issue

When the Express Migration Tool migrates Web Forms for Marketers module items, the tool is able to detect Web Forms for Marketers dictionary items, and automatically resolves the changes in them.

However, if you are using Web Forms for Marketers 2.4 rev. 150619 or Web Forms for Marketers 2.4 rev. 151103 on your source instance, these items will not be automatically resolved, because the approach of storing Web Forms for Marketers dictionary items was changed between these versions.

Therefore, the Express Migration Tool identifies all the Web Forms For Marketers dictionary items as either *New* or *Deleted*, however, you can still migrate all changes in the dictionary items by selecting them for migration.

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 *New* or *Deleted*, and the children items are identified as *Modified*.

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

Note
This known issue applies to all Sitecore and custom modules, except for the Web Forms for Marketers module.

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.
- Redeploy marketing definition items.
- 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 Downloads Page and the Sitecore Developer Network.

Editing a Patch File

To edit a patch file:

1. In a text editor, open each of the migrated configuration files and find the XML comments added by the tool that have a hyperlink to the Sitecore Downloads page – 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 configuration files are very important because they show the configuration file changes between the different versions of Sitecore.

2. In your browser, enter the URL that you found in the XML comments, open the configuration 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 Rebuild 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 between the source and target versions.

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 Downloads page.

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 restart the application pool:

- Launch Internet Information Services Manager (IIS) and stop and restart the application pool for the appropriate Sitecore target instance.

4.1.4 Validate and Add Languages

You must check that all of your system languages are present in the Sitecore backend, and if they are not, you must add them.

To validate that system languages are present in the Sitecore backend:

1. On the Sitecore Launchpad, click **Content Editor**.
2. In the content tree, navigate to the following path */sitecore/system/languages* and ensure that all of your relevant languages are listed.

To add a system language:

1. In the Content Tree, navigate to the */sitecore/system/languages* folder.
2. Select and right-click the *languages* folder, and in the context menu that appears, click **Insert**, **Language**.
3. Follow the instructions in the **Add a New Language** wizard.

4.1.5 Clean Up the Databases

When you clean up the databases, all item versions for languages that are not present in Sitecore are deleted.

Important
You must validate that all of your relevant system languages are present in Sitecore before you clean up the databases.

To clean up the databases:

1. On 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 to run the cleanup process, click **Clean**.

4.1.6 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. On 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**.

4.1.7 Deploy Marketing Definition

If you want to use the Sitecore Experience Marketing functionality, you must redeploy marketing definitions.

Note
If you do not plan to use the Sitecore Experience Database (xDB), you do not have to perform these steps.

To deploy 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**.

4.1.8 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. On 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** check box, 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. On 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 clear them on the **Selection** progressive tab.

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

A: No.

The third 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 then using the xDB Conversion Tool, upgrade to Sitecore 9.0.

Q: Does the tool automatically migrate Sitecore modules?

A: Except for the Web Forms for Marketers module, the Express Migration Tool does not automatically migrate modules.

The Express Migration Tool 3.0 supports the migration of the Web Forms for Marketers module, as well as custom modules.

Although version 3.0 of the tool does not support automatic migration of modules, 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
Except for the Web Forms For Marketers module, 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, ensure you clean up the tool's local databases or that you run a new instance of the tool.

- 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.