



# Kentico 8.1

1. Deploying websites	3
1.1 Precompiling websites	3
1.2 Content staging	4
1.2.1 Configuring content staging	5
1.2.2 Bi-directional content staging	7
1.2.3 Synchronizing the content	9
1.2.4 Automatic content synchronization	11
1.2.5 Staging large files	11
1.2.6 Using X.509 authentication	12
1.2.7 Content staging security	17
1.3 Deployment mode for virtual objects	17
1.4 Exporting and importing sites	18
1.4.1 Export folder structure	19
1.4.2 Export/import package directory browsing	20
1.4.3 Excluding files and folders from export	21
1.4.4 Exporting sites	21
1.4.5 Exporting objects	24
1.4.6 Exporting single objects	26
1.4.7 Importing a site or objects	26
1.4.8 Importing to web application projects	31
1.5 Publishing web application projects from Visual Studio	32

# Deploying websites

## Precompilation

Compile your web application into assemblies before deploying to improve the initial load times of your websites.

## Content staging

Break up the website's development, content creation, and deployment process. Set up different environments for various stages, for example: Development -> Testing -> Content editing -> Production.

## Export and Import

Export your website with all associated objects and settings, and then import the website to a different instance of Kentico.

## Microsoft Azure deployment

Deploy your sites to Microsoft Azure to ensure scalability, serve content from a CDN, and get access to other benefits of the cloud.

## Precompiling websites

When deploying a website, you have the option of [precompiling](#) the web project before you place it on a server. This compiles the project's code into assemblies, which provides several advantages:

- Faster initial response times - resources do not need to be compiled dynamically when they are requested for the first time.
- Source code protection - you can create a compiled version of the website without any accessible source code.



### Limitations of precompiled websites

Only use precompilation to deploy completed websites. Precompiled instances of Kentico have limited website development features.

You cannot edit, create or import virtual objects that contain ASP.NET markup and require compilation:

- [Page layouts](#) (ASCX type)
- [Transformations](#) (ASCX type)
- [Web part layouts](#)
- [Form layouts](#) (ASCX type)

If you wish to [import](#) physical files that require compilation (such as `cs`, `vb`, `aspx`, `ascx`), you need to run the import process on the original project, and then create the precompiled site again.

## Testing the website before compilation

You can check that your site's virtual objects compile correctly before starting the precompilation.

1. In the Kentico administration interface, open the **System** application.
2. Select the **Virtual objects** tab.
3. Click **Test virtual objects**.


The system compiles all virtual objects and adds them to the temporary files in the application's ASP.NET cache. This operation may take a long time, depending on the number of virtual objects in the system.

If the process finishes without errors, your website should run correctly after the precompilation.


## Precompiling web projects

1. In the Kentico administration interface, open the **System** application.
2. Select the **Virtual objects** tab.
3. Click **Store all virtual objects in file system**.

- This saves all virtual objects into the web project as physical files, so they can be included in the precompilation.
4. Open your web project in Visual Studio (using the **WebSite.sln** or **WebApp.sln** file).
  5. Right-click the Kentico project in the Solution Explorer and select **Publish Web Site**.
  6. Prepare a publish profile according to your requirements. See [How to: Deploy a Web Project](#) for details.
  7. On the **Settings** tab, expand **File Publish Options** and select **Precompile during publishing**.
    - Click **Configure** next to the check box and choose the compilation mode (**Allow precompiled site to be updatable**):
      - checked - compiles the code behind of all web forms and controls, but stores the markup files (.aspx, .ascx) in their original code form.
      - unchecked - the compilation also includes markup files. The website contains only empty "stub" .aspx/.ascx files without any code.

 We generally recommend using non-updatable sites for optimal performance.

- In the **Merge options** section, select the **Do not merge. Create a separate assembly for each page and control** option.
8. **Publish** the site.

 Alternatively, you can use command line tools to perform the precompilation and merging of assemblies:

- [ASP.NET Compilation Tool](#)
- [ASP.NET Merge Tool](#)

## Moving the database

You may need to copy the database to the server using the standard **backup/restore** procedure, since the compiled website cannot be used to install an SQL Server database. Other options are:

- Install a non-compiled website on the server first, go through the setup wizard and then replace the non-compiled files with compiled ones, while keeping the web.config file.
- Install Kentico locally and run the [database setup](#) against the remote SQL Server on the live server.

## Content staging

Staging allows you to easily transfer changes made to pages or objects in a Kentico instance on one server to a different instance on another server. You can also perform complete synchronization of all pages and objects. Staging is particularly useful if you need to synchronize multiple environments during development, for example:

**Development -> Testing -> Editing (staging) -> Live (production)**

All pages stored in the content tree of websites can be synchronized. However, not all objects in Kentico support synchronization. See the [What can be synchronized](#) section for more information.

## Setting up staging

[Configuring content staging](#) - to allow synchronization, you need to configure the Kentico instances that you wish to connect.

[Bi-directional content staging](#) - explains how to configure servers to be both target and source servers at the same time. Allows transferring of changes in both directions.

[Using X.509 authentication](#) - learn how to install your own X.509 certificates or how to use our sample certificates for the X.509 type of staging service authentication.

[Staging large files](#) - Kentico supports staging of **physical files** stored as [Page attachments](#) and in [Media libraries](#). Learn how to set the maximum size of synchronized files.

[Content staging security](#) - learn about the permissions that roles need to perform synchronization actions.

## Synchronization

[Synchronizing the content](#) - explains how to transfer synchronization tasks to other servers.

[Automatic content synchronization](#) - set up the system to perform synchronization automatically.

## Other

[Exporting and importing sites](#) - you can also manually transfer your content to other instances of Kentico using the export and import features.

## What can be synchronized

Content staging **supports** synchronization of the following data:

- [Page data](#) - pages in the website content tree. Timeout defined for pages that are part of a workflow is not synchronized for consistency reasons.
- [Page file attachments](#) - if a page contains attachments or file fields, the files are synchronized together with the page.
- [Page relationships](#) - relationships between pages are synchronized if the relationship type and both pages exist on the target server. Synchronization is NOT supported for relationships between pages on different sites.
- [Workflow process](#) - only published page versions are synchronized to the target server and both servers need to have the same workflow schemas defined.
- [Custom tables](#) and their data.
- [Media libraries](#) and the files and folders in them.
- [ACLs \(page-level permissions\)](#)
- **Global objects** - all global objects except for the exceptions listed below.

Content staging **does not support** synchronization of the following data:

- [Abuse reports](#)
- [Accounts](#)
- [Activities](#)
- [Blog comments](#) (blogs and blog posts are synchronized as pages)
- [Booking event attendees](#)
- [Contacts](#) (the state of contacts in [automation processes](#) is also not synchronized)
- [Custom modules](#)
- [Event log](#)
- [Export history](#)
- [Form data](#) (the actual form objects are synchronized)
- [Forum posts](#) (forum objects are synchronized)
- [Friends](#)
- [Message board messages](#) (message board objects are synchronized)
- [Personal messages](#)
- [Personas](#)
- Physical files associated with global objects (for example web part source file, form control files, ASPX page template files)
- [Web templates](#)

### Staging and ID values

Content staging cannot ensure that objects and pages have the same ID values after being transferred to a different environment. However, the synchronization process preserves [GUID](#) values. Use GUID fields if you need to identify pages or objects across multiple staging environments.

## Configuring content staging

Configuration of content staging consists of the following parts:

1. [Source server configuration](#) - configuration of the server from which changes will be transferred to the target servers.
2. [Target server configuration](#) - configuration of the server to which changes will be transferred from the source servers.
3. [Configuring servers for staging macro expressions](#)

You also need to ensure that all instances use the same settings (page types, templates, web parts ...), code files, and that both servers use the same version of Kentico.

## Configuring the source server

To configure a Kentico instance as a source server, you first need to enable logging of content staging tasks. Open the **Settings** application, and adjust the settings in the **Versioning & synchronization -> Staging** category.

- **Log content changes** - if enabled, synchronization tasks are automatically logged when content (a page) is modified.
- **Log data changes** - if enabled, synchronization tasks are automatically logged when custom tables data is modified.
- **Log object changes** - if enabled, synchronization tasks are automatically logged when supported objects are modified.
- **Log staging changes** - if enabled, synchronization tasks are logged for changes made by synchronization from another server to this server. See [Bi-directional content staging](#) for more details.
- **Log export tasks** - if enabled, synchronization tasks are logged when an object is deleted (incremental update support).

With these settings enabled, the system logs all changes to the corresponding content as staging tasks. These tasks can then be transferred to the target servers and performed there to synchronize the content.

## Defining target servers

To specify the target servers, open the **Staging** application and select the **Servers** tab.

Add servers by clicking **New server**. Set the following properties for each staging server:

Staging server properties	Description
Display name	Name of the server displayed to users in the administration interface.
Code name	Unique identifier of the server.
Server service URL	Enter the root URL of the target Kentico instance, i.e. the protocol, domain, and virtual directory (if necessary). For example: <i>http://www.targetserver.com</i>  The system automatically builds the full service URL by appending <i>/CMSPages/Staging/syncserver.asmx</i> to the value.  Click <b>Check Server availability</b> to confirm whether the entered URL is available.
Enabled	If checked, the system generates synchronization tasks for the server. You can temporarily disable the server by disabling this flag, for example during server maintenance.
Server authentication	Server authentication settings. Configure according to the settings of the given target server (described below). The default user name is <b>admin</b> and the default password is <b>pass</b> .  If you want to use X509 authentication, see <a href="#">Using X.509 authentication</a> .



**Note:** The system only logs staging tasks if at least one target server is created and enabled.

## Configuring the target servers

On the target server, the staging service is disabled by default. You need to adjust the following settings in **Settings -> Versioning & synchronization -> Staging**:

- **Enable staging service** - enables the staging service for the given site.
- **Staging service authentication** - we recommend starting with *Username and password* authentication first, testing the synchronization, and then optionally configuring the site for *X509* certificates.
  - *USERNAME* - username/password authentication (fast, recommended for data without high security requirements).
  - *X509* - X509 certificate authentication (more secure, slower, requires certificates). See: [Using X.509 authentication](#)
- **Staging service username** and **password** - username and password for the *Username and password* authentication.
- **Server key ID** and **Client key ID** - certificate keys for the *X509* authentication.

## Configuring servers for staging macro expressions

The system uses signatures to ensure the security of **macro expressions**. Macro signatures contain the user name of the macro's author and a hash of the given expression.

The hash function used to create the signatures appends a **salt** to the input. The salt value depends on the configuration of individual applications, so the signatures are only valid in the environment where the macros were originally saved. To allow macros to work correctly on all staging servers, you need to assign the same custom hash salt to all servers:

- Set the **CMSHashStringSalt** key in the *appSettings* section of the web.config file to the same value on all staging servers. You can use any string as the value, but the salt should be random and at least 16 characters long. For example, a randomly generated GUID is a strong salt:

```
<add key="CMSHashStringSalt" value="e68b9ad6-a461-4707-8e3e-ece73f03dd02" />
```

The best option is to set the hash salt value before you start creating content for your website. Changing the salt causes all current hash values to become invalid. To fix existing macro expressions in the system after changing the hash salt, you need to re-sign the macros. See [Working with macro signatures](#) for more information.



**Warning:** In addition to macro signatures, the system uses the **CMSHashStringSalt** value for other hash functions. Changing the

hash salt on a website that already has defined content may break dialog links and images on your website. If you encounter such problems, you need to re-save the given content (the system then creates the hashes using the new salt).

## Customizing staging via event handlers

Developers can use [event handlers](#) to modify or extend the content staging functionality. See the following pages for more information:

- [Reference - Staging events](#)
- [Excluding content from staging and integration](#)
- [Automatically synchronizing staging and integration tasks](#)

## Bi-directional content staging

You can set up bi-directional content staging between servers — the servers are not exclusively source or target servers, and can transfer changes both ways. The advantage of bi-directional configurations is that you can make changes not only on the source server, but also on the other servers. The changes are then transferred to the rest of the servers.

There are two types of bi-directional content staging, depending on the number of servers between that you want to synchronize:

- [Simple](#) - used for synchronization between 2 servers only
- [Advanced](#) - for synchronization between more than 2 servers

### Simple bi-directional staging

If you want to perform content staging bi-directionally between **two servers**, you need to configure both servers as a source and a target server at the same time, as described in [Configuring content staging](#).

Additionally, you need to prevent synchronization loops. Perform the following on both servers:

1. Open the **Settings** application.
2. Select the **Versioning & Synchronization -> Staging** category.
3. Disable the **Log staging changes** setting.
4. Click **Save**.

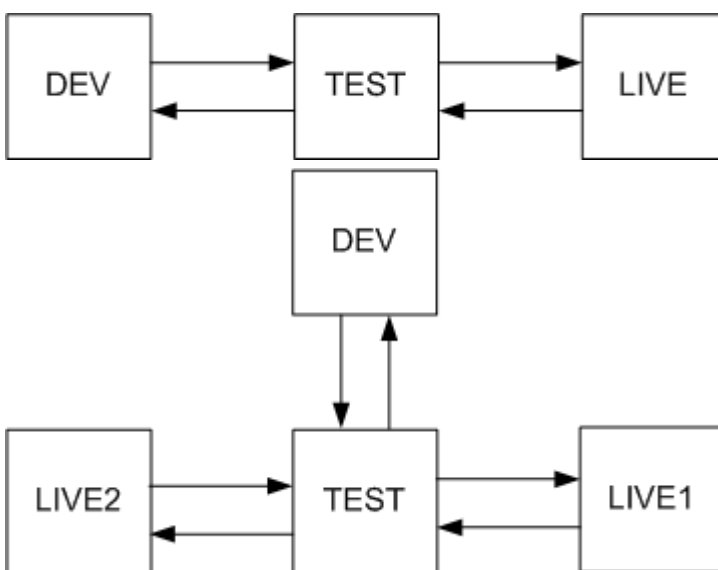
This ensures that the servers do not attempt to synchronize changes that are the result of staging from the other server.

### Advanced bi-directional staging

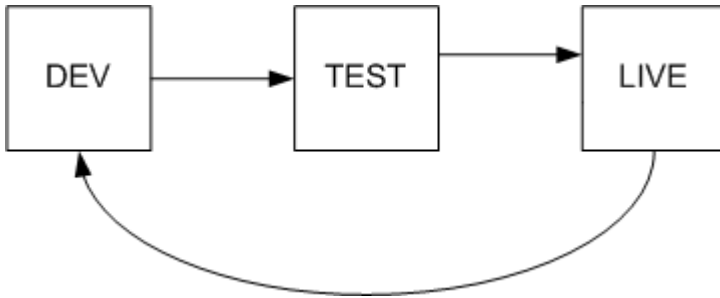
Bi-directional content staging is also possible on more than two servers. In this case, the data of each transferred staging task contains a list of servers where the task was already processed. This prevents redundant staging tasks from being logged for servers where the changes have already been performed.

The rule is that you need **only one path between any two servers in both directions**. The following figures show examples of supported server topologies. The rectangles represent servers, and the arrows indicate the flow of content staging synchronization tasks.

#### Star topology

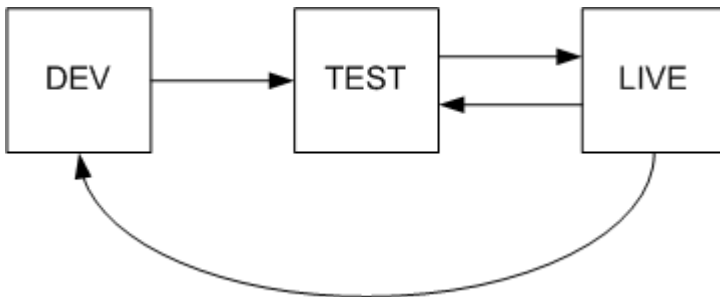


#### Circle topology



Other topologies are not supported. The following diagram shows an unsupported environment — there are two paths from the LIVE server to the TEST server, which breaks the rule.

#### Unsupported topology (example)



### Configuring advanced bi-directional staging

The following steps describe how to configure bi-directional staging for three servers using a star topology — a development, test and live server.

For all servers that you want to include in the bi-directional staging:

1. Open the **Settings** application.
2. Select the **Versioning & Synchronization -> Staging** category.
3. [Configure](#) the settings for both source and target staging.

#### Configuring the Development server

1. Edit the web.config file of the application.
2. Add the following key into the *appSettings* section:

```
<add key="CMSStagingServerName" value="Dev" />
```

3. In the Kentico administration interface, open the **Staging** application.
4. Select the **Servers** tab and click **New server**.
5. [Configure](#) the new server to target the **Test** server.
  - Set the **Server code name** to **Test**, which is the value that you will use to identify the test server in the web.config.

#### Configuring the Test server

1. Edit the web.config file of the application.
2. Add the following key into the *appSettings* section:

```
<add key="CMSStagingServerName" value="Test" />
```

3. In the Kentico administration interface, open the **Staging** application.
4. Select the **Servers** tab and click **New server**.
5. [Configure](#) the new server to target the **Dev** server.
  - Set the **Server code name** to **Dev**, which identifies the development server.
6. Create another server.



7. [Configure](#) the new server to target the **Live** server.
  - Set the **Server code name** to **Live**, which is the value that you will use to identify the live server in the web.config.

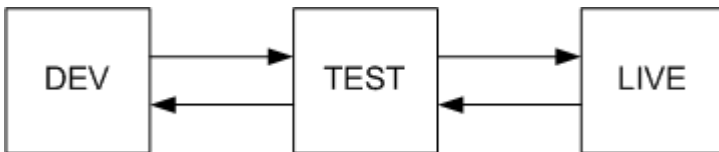
### Configuring the Live server

1. Edit the web.config file of the application.
2. Add the following key into the *appSettings* section:

```
<add key="CMSStagingServerName" value="Live" />
```


3. In the Kentico administration interface, open the **Staging** application.
4. Select the **Servers** tab and click **New server**.
5. [Configure](#) the new server to target the **Test** server.
  - a. Set the **Server code name** to **Test**, which identifies the test server.

You have now set up bi-directional staging between three servers.



## Synchronizing the content

Once you [configure content staging](#), the system tracks all changes made to pages and objects. You can view the synchronization tasks representing the changes in the **Staging** application.

 Before you synchronize content between two servers, we recommend setting the same **Code name** for the source and target sites. To change the value, open the **Sites** application and **Edit** (✎) the appropriate sites.

The interface for viewing changes and performing synchronization is divided into the following tabs:




### All tasks tab

On the **All tasks** tab, you can see a list of all content staging tasks, i.e. all changes made to the system that can be synchronized on the target server.



















Using the **Server** drop-down, you can choose the target server that you want to synchronize. By choosing **(all)**, you perform synchronization for all available target servers. Then you can perform one of the following actions using the buttons at the bottom:

- **Synchronize selected** - performs synchronization for all selected tasks on the target server.
- **Synchronize all** - performs synchronization for all listed tasks on the target server. If you made any changes to content on the target server in the meantime, these changes will be overwritten.
- **Delete selected** - deletes all selected tasks for the target server.
- **Delete all** - deletes all listed tasks on the target server.

You can also perform the following actions separately with specific staging tasks:

-  **View** - opens a new window with detailed information about the staging task.
-  **Synchronize** - performs the synchronization task on the target server.
-  **Delete** - deletes the synchronization task from the list.

All tasks Server: (all) ▼

	Actions	Task title	Task type	Task time ▲	Result
<input type="checkbox"/>	  	Update Page template 'Corporate Site - Simple text'	UPDATEOBJ	3/1/2014 8:29:24 AM	
<input type="checkbox"/>	  	Update document Web Design	UPDATEDOC	3/1/2014 8:29:31 AM	
<input type="checkbox"/>	  	Update Form 'Contact Us'	UPDATEOBJ	3/1/2014 8:30:25 AM	
<input type="checkbox"/>	  	Delete Custom table item 'Sample table - Foo text'	DELETEOBJ	3/1/2014 8:31:29 AM	
<input type="checkbox"/>	  	Create Custom table item 'Sample table - Sample text 4'	CREATEOBJ	3/1/2014 8:31:38 AM	
<input type="checkbox"/>	  	Update Forum 'Website forums'	UPDATEOBJ	3/1/2014 8:31:48 AM	

Items per page: 15 ▼

Synchronize selected
Synchronize all
Delete selected
Delete all

## Pages tab

On the **Pages** tab, you can see logged changes made to pages. By clicking the website root in the content tree on the left, you can view a list of all changes (synchronization tasks) made to all the pages of your website. By clicking individual pages, you can view only the changes made to the given page.

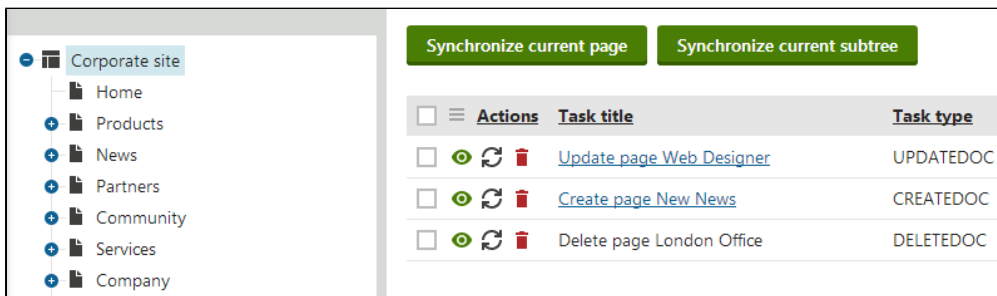
You can perform the same actions as on the **All tasks** tab, as described in the [All tasks tab](#) section.

You can also perform the following manual actions. These actions are manual because they are not related to the listed tasks and they can be performed even if there are no synchronization tasks logged:

- **Run complete synchronization** - performs complete synchronization of all pages in the content tree.
- **Synchronize current page** - synchronizes the currently selected page.
- **Synchronize current sub-tree** - synchronizes all pages in the selected sub-tree.

The following types of tasks are logged for pages. You can see the type in the **Task type** column:

- **CREATEDOC** - the page was created.
- **UPDATEDOC** - the page was modified.
- **DELETEDOC** - the page was deleted.
- **DELETEALLCULTURES** - all cultural versions of the page were deleted.
- **PUBLISHDOC** - the page was published.
- **ARCHIVEDOC** - the page was archived.
- **REJECTDOC** - the page was rejected.
- **MOVEDOC** - the page was moved to another location in the content tree.



The screenshot shows the 'Pages' tab interface. On the left is a content tree for 'Corporate site' with folders: Home, Products, News, Partners, Community, Services, and Company. On the right, there are two buttons: 'Synchronize current page' and 'Synchronize current subtree'. Below these is a table of tasks:

<input type="checkbox"/>	Actions	Task title	Task type
<input type="checkbox"/>		<a href="#">Update page Web Designer</a>	UPDATEDOC
<input type="checkbox"/>		<a href="#">Create page New News</a>	CREATEDOC
<input type="checkbox"/>		Delete page London Office	DELETEDOC

## Data tab

On the **Data** tab, you can see the logged changes for the data of [custom tables](#).

You can perform the same actions as on the **All tasks** tab, as described in the [All tasks tab](#) section.

You can also perform the following manual actions. These actions are manual because they are not related to the listed tasks and they can be performed even if there are no synchronization tasks logged:

- **Run complete synchronization** - performs complete synchronization of all data in all custom tables.
- **Synchronize current subtree** - synchronizes all data in the selected custom table.

The following types of tasks are logged for custom tables data. You can see the type in the **Task type** column:

- **CREATEOBJ** - a new item was added to a table.
- **UPDATEOBJ** - an item in a table was updated.
- **DELETEOBJ** - an item in a table was deleted.



Staging tasks for custom table items are logged based on their *ItemGUID* columns. Changes made to items that do not have this column (typically custom tables imported from older versions of Kentico) are not logged.

You can edit custom tables in the **Custom tables** application. The application displays a warning message with a link that lets you generate GUIDs for custom tables.

## Objects tab

On the **Objects** tab, you can view logged changes made to objects in the system.

The tree allows you to filter the tasks according to object types. The **Website** category contains object changes related to the currently selected site, and the **Global objects** category contains changes of global objects.

You can perform the same actions as on the **All tasks** tab, as described in the [All tasks tab](#) subsection.

When you select an object category, you can click **Synchronize current subtree** to synchronize all objects in the given category.

The system logs the following **Task types** for objects:

- **CREATEOBJ** - a new object was created.
- **UPDATEOBJ** - the object was modified.

- **DELETEOBJ** - the object was deleted.
- **ADDTOSITE** - the object was assigned to a site; applicable only to site-related objects.
- **REMOVEFROMSITE** - the object was removed from a site; applicable only to site-related objects.

The following types of tasks are logged for folders in media libraries:

- **CREATEFOLDER** - a folder was created.
- **RENAMEFOLDER** - the folder was renamed.
- **COPYFOLDER** - the folder was copied.
- **MOVEFOLDER** - the folder was moved.
- **DELETEFOLDER** - the folder was deleted.

**i** Global metadata changes to objects such as page types, custom tables and system classes produce staging tasks for all staging servers of all sites. We recommend synchronizing such changes at the same time on all servers to prevent overwriting of data (you may lose data by synchronizing older tasks later).

You can use the `<add key="CMSStagingTreatServerNamesAsInstances" value="true" />` web.config key to make sure that once the global task is synchronized, it is deleted from all other servers with the same name. The default value is false since staging can use multiple target instances targeted with the same names.

Actions	Task title	Task type	Task time	Result
<input type="checkbox"/>	Update Page template 'Corporate Site - Simple text'	UPDATEOBJ	3/1/2014 8:29:24 AM	
<input type="checkbox"/>	Update Form 'Contact Us'	UPDATEOBJ	3/1/2014 8:30:25 AM	
<input type="checkbox"/>	Update Forum 'Website forums'	UPDATEOBJ	3/1/2014 8:31:48 AM	

## Automatic content synchronization

You can set up the system to automatically synchronize changes from the staging environment to the target server, without waiting for approval from administrators.

### Scheduling synchronization of page changes

To regularly synchronize page staging tasks, configure the **Content synchronization** scheduled task:

1. Open the **Scheduled tasks** application.
2. Select the **Site** whose pages you want to synchronize.
3. Edit () the **Content synchronization** task.
4. Select the **Task enabled** check box.
5. Schedule the task (set the Period, Start time, Every, Between, Days properties).
6. Click **Save**.

**i** The scheduled task **only synchronizes page changes**. You cannot schedule synchronization for changes of custom table data or objects.

### Automatically synchronizing tasks using event handlers

Developers can leverage the API and event handlers to automatically synchronize any type of staging tasks according to custom requirements.

See [Automatically synchronizing staging and integration tasks](#) to learn more.

## Staging large files

Kentico allows you to limit the size of files that are synchronized by [staging](#). Files that support staging include:

- [Page attachments](#)
- [Files in Media libraries](#)

You can set a size limit to prevent the system from using too many resources by synchronizing files that are too large. You can also increase the limits if you need to stage files that are larger than is allowed by default.





The system only logs staging tasks for files when the changes are performed via the Kentico administration interface. If you make changes directly in the file system, for example upload or update files using FTP, the changes are not logged.

If you make a change to a file via the administration interface and then update the file using FTP, the current file (the one updated using FTP) will be transferred to the target server, even if the staging task was created before the file upload. This happens because the system loads the binary data of the files at the time of synchronization, not when the synchronization task is logged.

## Changing the maximum content and request size

One way to restrict staging of files is to adjust the limits of the maximum request size for the staging service:

1. On the **target server**, edit the **web.config** file in the **CMS** folder of the web project.
2. Add a **<location>** element for the staging service into the **<configuration>** section and set the following attributes:
  - **maxRequestLength** (**<httpRuntime>** element under **<system.web>**). The value is in **kiloBytes**.
  - **maxAllowedContentLength** (**<security>/<requestFiltering>/<requestLimits>** element under **<system.webServer>**). The value is in **Bytes**.

```
<location path="CMSPages/Staging">
  <system.web>
    <httpRuntime executionTimeout="2400" maxRequestLength="4096" />
  </system.web>
  <system.webServer>
    <security>
      <requestFiltering>
        <requestLimits maxAllowedContentLength="4096648" />
      </requestFiltering>
    </security>
  </system.webServer>
</location>
```

3. Save the web.config file.

Synchronizing files larger than the set value results in failure (the target server returns an *HTTP status 404: Not Found* error).



**Important:** The binary data of files only forms a part of the synchronization requests. You may need to set a higher maximum value than the file size that you wish to allow.



You can also configure the limits on the **source server** by setting the **maxRequestLength** and **maxAllowedContentLength** attributes inside the main **<system.web>** and **<system.webServer>** sections of the web.config (NOT within a specific **<location>** section)

This affects all types of requests, not just the staging service.

## Limiting maximum media library file size

To limit the maximum size of files staged from **media libraries**:

1. On the **source server**, edit the **web.config** file in the **CMS** folder of the web project.
2. Add the **CMSMediaFileMaxStagingSize** key into the **<appSettings>** section of the web.config. The value is in **kiloBytes**.

```
<add key="CMSMediaFileMaxStagingSize" value="102400" />
```

3. Save the web.config file.

The system does not synchronize media library files above the limit — the synchronization only transfers the database definitions of the media objects without the files. You can find records of files not being synchronized in the **Event log**.

## Using X.509 authentication

You can choose between two types of **staging service** authentication – Username and password authentication, and **X.509**. If you want to use X.509 authentication, you need to install your own certificates or use the provided sample certificates.

X.509 authentication is slower and more difficult to configure, but also more secure.

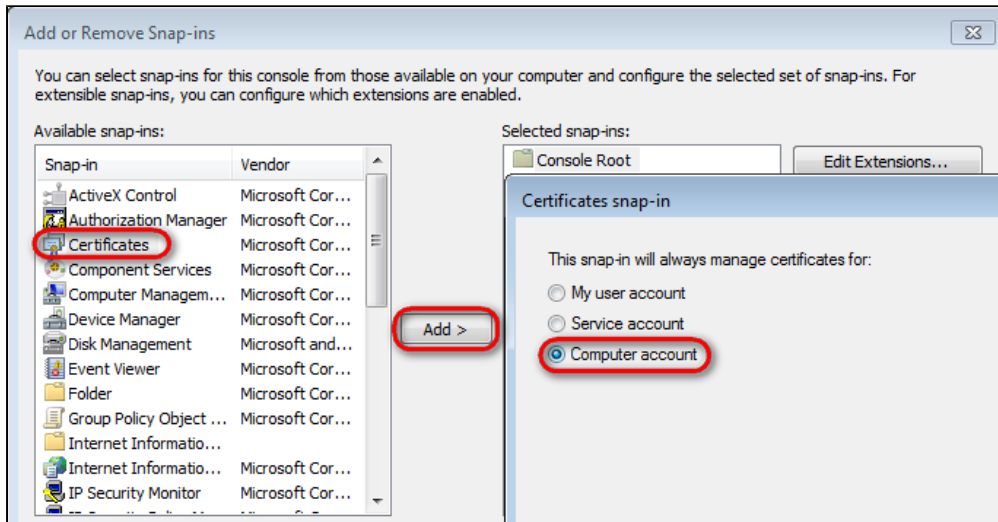
## Using the sample certificates

Kentico contains sample client and server private certificates. To install the sample certificates, perform the following tasks on the source and target servers.

### Installing the server certificate

To install the server certificate:

1. In Windows, type **mmc** in the Start menu search box press Enter.
2. In the console window, choose **File -> Add/Remove Snap-in**.
3. Select **Certificates** and click **Add**.
4. Choose **Computer account** and click **Next**.

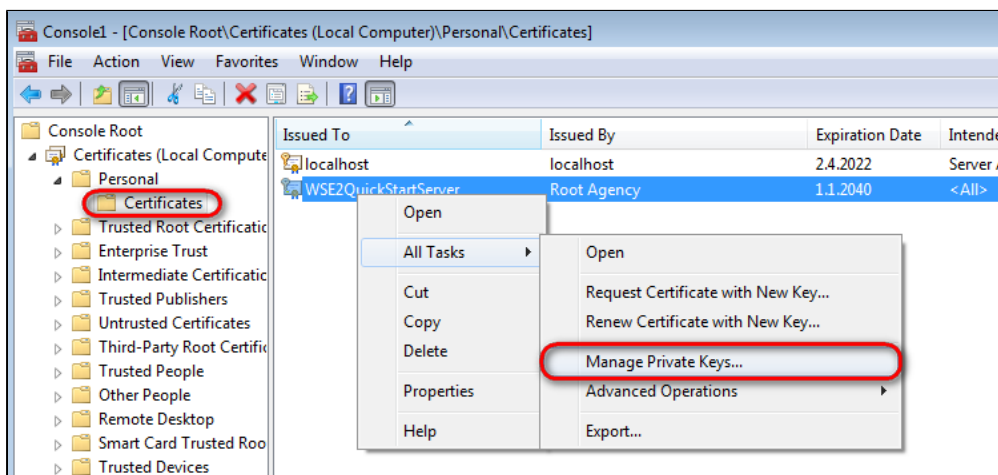


5. Choose **Local computer (the computer this console is running on)** and click **Finish**.
6. Close the **Add or Remove Snap-ins** window by clicking **OK**.
7. Unfold **Certificates (Local Computer)** under the console root, right-click **Personal** and choose **All Tasks -> Import**.
8. Import the *Server private.pfx* file located in **C:\Program Files\Kentico\<version>\SampleCertificates**.
9. Enter the following password for the sample certificate: **wse2qs**.
10. Leave the other settings and finish the **Certificate Import Wizard**.

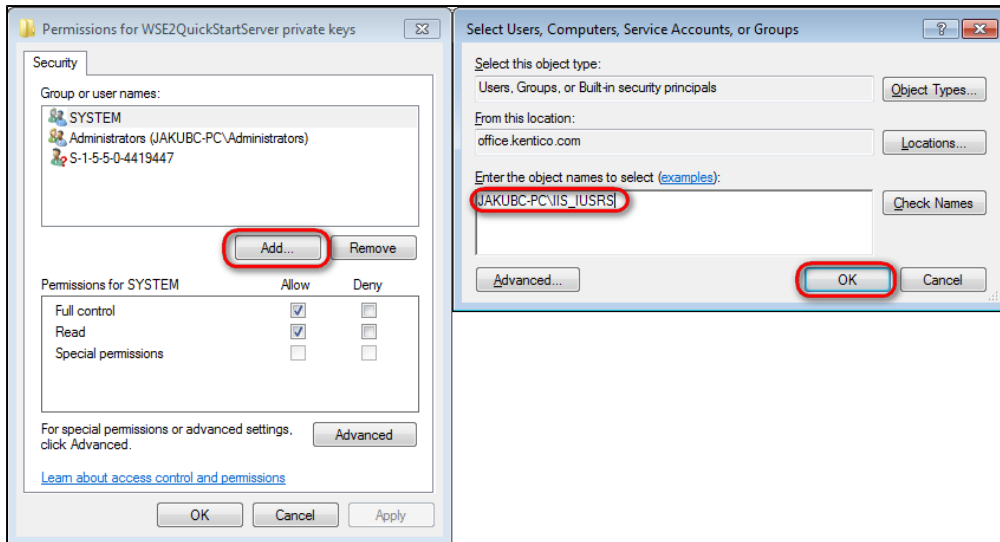
### Granting the Read permissions for the Server certificate

Grant the **Read** permission to the certificate file for the ASP.NET account.

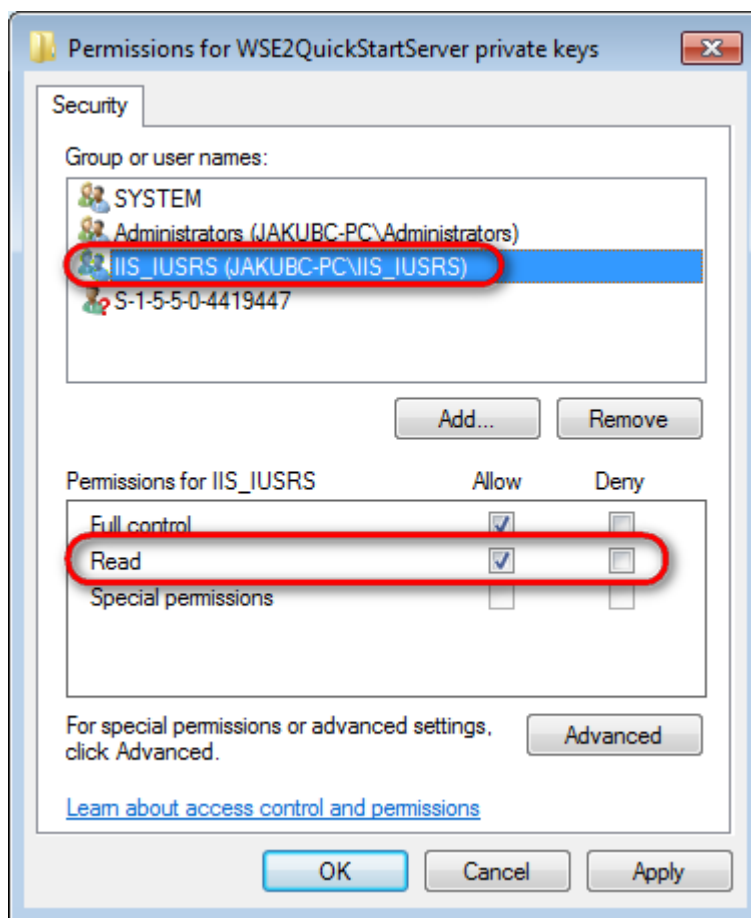
1. Right-click the imported *WSE2QuickStartServer* certificate and choose **All tasks -> Manage private keys**.



2. Click **Add...**, fill in the name of the account.
3. Click **OK**.



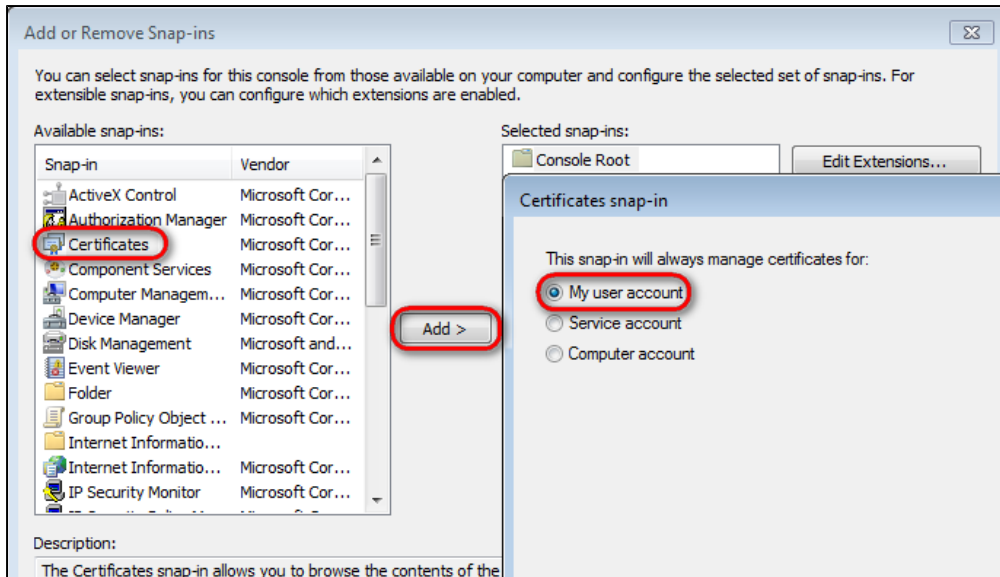
4. Make sure the account's permission is set to **Read - Allow** and click **OK**.



### Installing the client certificate

To install the client certificate:

1. In Windows, type **mmc** in the Start menu search box press Enter.
2. In the console window, choose **File -> Add/Remove Snap-in**.
3. Choose **Certificates** and click **Add**.
4. Choose **My user account** and click **Finish**.



5. Close the **Add or Remove Snap-ins** window by clicking **OK**.
6. Expand **Certificates - Current User** under the console root, right-click **Personal** and choose **All Tasks -> Import**.
7. Import the *Client private.pfx* file located in **C:\Program Files\Kentico\<version>\SampleCertificates**.
8. Enter the following password for the sample certificate: **wse2qs**.
9. Leave the other settings and finish the **Certificate Import Wizard**.

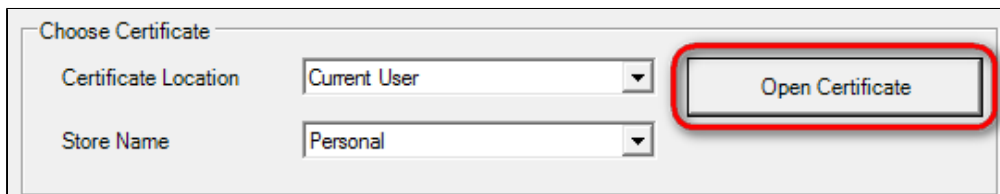
### Granting the Read permissions for the Client certificate

You need to grant the **Read** permissions for the certificate file to the ASP.NET account. We recommend that you install the **WSE 3.0** tool:

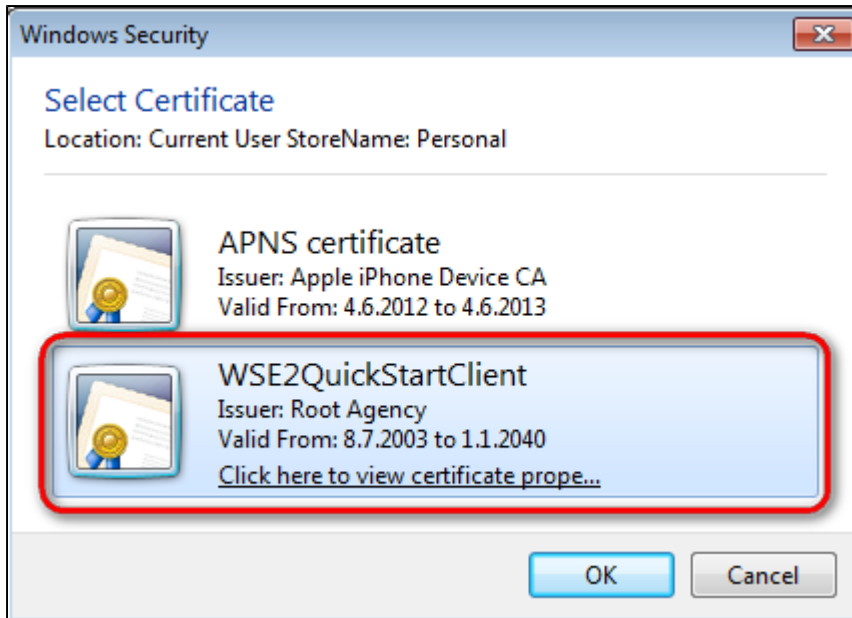
1. [Download the WSE 3.0 installation file](#).
2. Run the downloaded file.
3. In the Setup Type choose the **Administrator** type of installation.
4. Install the utility.

When the utility is installed, you can use it to grant the Read permissions to the certificate file:

1. Run the **Certificates tool** from Start -> Microsoft WSE 3.0.
2. Choose **Current User** in the Certificate Location field.
3. Choose **Personal** in the Store Name field.
4. Click **Open Certificate**.



5. Choose the *WSE2QuickStartClient* certificate and click **OK**.



6. Click **View Private Key File Properties...**
7. Switch to the **Security** tab and click **Edit...**
8. Click **Add...**, fill in the name of the account and click **OK**.
9. Make sure the account's permission is set to **Read - Allow** and click **OK**.

### **Sample certificates**

Using the sample certificates is not secure and also very slow. We highly recommended using your own certificates issued by a certification authority.

## Using your own certificates


If you are using your own certificates (highly recommended), you need to obtain **Client key ID** and **Server key ID** values for your certificates.

To get the IDs, we recommend that you install and use the **WSE 3.0** tool:

1. Download the [WSE 3.0 installation file](#).
2. Run the downloaded file.
3. In the Setup Type choose the **Administrator** type of installation.
4. Install the utility.

When the utility is installed, you can use it to obtain the values for your certificates:

1. Run the **Certificates tool** from Start -> Microsoft WSE 3.0.
2. Select the **Certificate Location** and **Store Name** for your certificate.
3. Click **Open certificate** and select either the client or server certificate.
4. In the **Key identifiers** group you can now see the certificate key. Use the **Windows key identifier (Base64 Encoded)**.

 If you encounter problems with content staging when using SLL (X.509), try adding the following key to your *web.config* file:

```
<add key="CMSStagingAcceptAllCertificates" value="true" />
```

This key ensures that all certificates will be accepted. If set to false, only certificates issued by a certification authority will be accepted.

## Configuring staging for the use of certificates

Now that you have installed and configured the certificates, adjust the staging settings in Kentico to use the certificates for authentication.

### Target server

On the target server, change the staging service authentication type:

1. Open the **Settings** application.
2. Select the **Versioning & Synchronization** -> **Staging** category.
3. Set the **Staging service authentication** setting to **X.509**.



4. Fill in the **Client key ID** and **Server key ID**.
5. Click **Save**.

### Source server

On the source server, adjust the settings of the target servers:

1. Open the **Staging** application.
2. Select the **Servers** tab.
3. Edit (✎) the target servers.
4. Change the **Server authentication** to **X509** and copy the **Client** and **Server key ID's** from the target server.
5. **Save** the configuration.

The staging service now uses certificates during authentication.

## Content staging security

You can control access to [content staging](#) in the **Permissions** application.

- **Site:** select your site
- **Permissions for :** Module, Staging

In the permission matrix, you can grant the following permissions to the available [roles](#):

- **Manage servers** - allows members of the roles to manage target server configurations on the **Servers** tab of the **Staging** application.
- **Manage all tasks** - allows access to the **All tasks** tab of the **Staging** application, and synchronization/management of synchronization tasks.
- **Manage page tasks** - allows access to the **Pages** tab of the **Staging** application, and synchronization/management of synchronization tasks.
- **Manage object tasks** - allows access to the **Objects** tab of the **Staging** application, and synchronization/management of synchronization tasks.
- **Manage data tasks** - allows access to the **Data** tab of the **Staging** application, and synchronization/management of synchronization tasks.

Role	Manage servers	Manage all tasks	Manage page tasks	Manage object tasks	Manage data tasks
Authenticated users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CMS Basic users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CMS Community administrators	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CMS Designers	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CMS Desk Administrators	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
CMS Editors	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## Deployment mode for virtual objects

Kentico provides a way to store virtual objects in the file system in addition to the database. This can be useful if you need to prepare your site for deployment, or if you want to manage the code of objects [using a source control system](#).

Virtual objects that require compilation include the following:

- [Form layouts](#) (ASCX type)
- [Page layouts](#) (ASCX type)
- [Transformations](#) (ASCX type)
- [Web part layouts](#)

To prepare a website for deployment to another system:

1. Open the **System** application.
2. Select the **Virtual objects** tab.
3. Click **Store all virtual objects in file system**.
  - The system saves all virtual objects (alternative form layouts, form layouts, page layouts, page templates, transformations and web part layouts) on your disk.
  - The target folder is `~/CMSVirtualFiles`.
4. Compile the files in Visual Studio.
  - At this time, you can still edit the code of objects in the Kentico user interface, but any changes will require you to compile the files again.

When deployment mode is **OFF**:

- The storage location of the virtual objects depends on the settings of the **Source control** options.

When deployment mode is **ON**:

- All objects that require compilation are moved into the file system on the local disk. The target folder is `~/CMSVirtualFiles`.
- You can edit the code of objects in the user interface. Changes are saved into the files immediately.
- You can edit the object files in Visual Studio or another external editor.
- To move objects back into the database, click **Store all virtual objects in database** (deletes the files on the disk).
- You can configure the **Source control** options for objects that do not require compilation (Web part containers and CSS stylesheets).

General	<h3>Deployment</h3> <p>The deployment mode is <b>on</b>, all virtual files are stored in the file system.</p> <p><b>Store all virtual objects in database</b></p>
E-mail	
Files	
Virtual objects	<h3>Source control</h3> <p>Source control is disabled for virtual objects in Deployment mode. Turn t</p> <p><input checked="" type="checkbox"/> Alternative form layouts</p> <p><input checked="" type="checkbox"/> Form layouts</p> <p><input checked="" type="checkbox"/> Page layouts</p> <p><input checked="" type="checkbox"/> Page template layouts</p> <p><input checked="" type="checkbox"/> Transformations</p> <p><input checked="" type="checkbox"/> Web part layouts</p> <p><input type="checkbox"/> CSS stylesheets</p> <p><input type="checkbox"/> Web part containers</p> <p>Click 'Apply changes' button to store selected objects in file system. Clic</p> <p><b>Apply changes</b>      <b>Synchronize changes to database</b></p>
DB separation	
Macros	



**Note:** Do not [apply hotfixes](#) while deployment mode is on. Disable deployment mode before you start the hotfix procedure, and re-enable it once the hotfix is applied.

## Deployment mode on web application projects

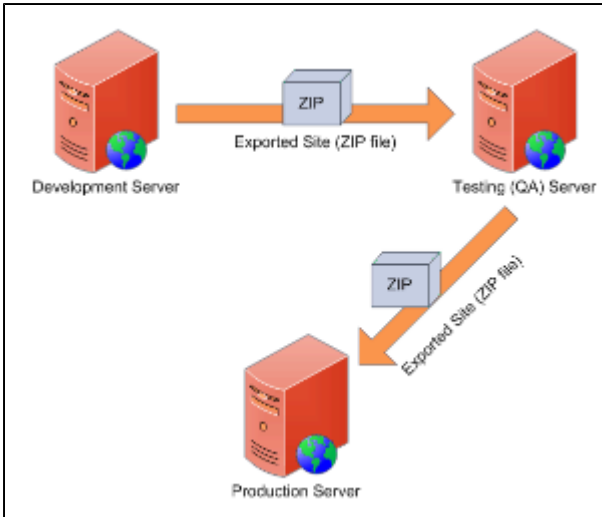
If your project was installed as a web application, you need to perform additional steps after enabling deployment mode:

1. Edit your project in Visual Studio.
2. Click **Show all files** at the top of the Solution Explorer.
3. Right-click the **CMSVirtualFiles** folder and select **Include in Project**.
4. Right-click the **CMSVirtualFiles\WebPartLayouts** folder and select **Convert to Web Application**.
5. Build the **CMSApp** project.

The deployed files are now integrated in your web application project. Objects that have code behind files ([Web part layouts](#)) are converted to the web application format with designer files.

## Exporting and importing sites

You can export and import website content and settings from one Kentico instance to another. You can use the import and export features to manually move websites or selected objects between the development, testing and production (live) servers.



## General

[Export folder structure](#) - learn how the folder structure of your web project is affected during export and import.

[Export/import package directory browsing](#) - protect your exported packages against file enumeration attacks.

[Excluding files and folders from export](#) - you can leave out specific files and folders from the export.

## Export

[Exporting sites](#) - export entire websites, including their settings and related objects.

[Exporting objects](#) - export only selected objects (web parts, page types, page templates, etc.).

[Exporting single objects](#) - you can export certain objects individually to create backups (for example CSS stylesheets, E-mail templates or Page layouts).

## Import

[Importing a site or objects](#) - import exported website packages or object packages.

[Importing to web application projects](#) - you need to take additional steps when importing into Kentico instances installed in the web application format.



### Export and ID values

The system cannot ensure that pages and objects have the same ID values after being exported from one environment to another. However, the export process preserves GUID values. Use GUID fields if you need to identify pages or objects that have been transferred across multiple environments.

## Export folder structure

Kentico instances organize all files in the single **CMS** folder, even if you manage multiple websites in one installation.

The following list describes the main folders and how they are affected during the import and export.

Color	Meaning
red	System folders. Do not make any changes or place your files here unless you want to modify the administration interface.
blue	Folders for custom files, part of the export package.

green	Folders for custom files, part of the export package. You may need to create these folders manually.
black	Service folders (import files, import/export).

- **Admin**
- **App\_Browsers**
- **App\_Code** (or **Old\_App\_Code** if you installed Kentico as a web application)
  - CMSModules\**<module\_name>** (folders of custom modules)
  - **Global** (exports with any site, needs to be created manually; the folder is exported if the 'Export global folders' option is checked during the export process)
  - **<site code name>** (exports with given site, needs to be created manually; the folder is exported if the 'Export site folders' option is checked during the export process)
- **App\_Data**
  - CMSModules\**<module\_name>** (folders of custom modules)
- **App\_Themes**
  - **<stylesheet name>** (all folders related to stylesheets assigned to or used on the website)
- **App\_WebReferences**
- **bin**
- **ClientBin**
- **CMSAdminControls**
- **CMSAPIExamples**
- **CMSDesk**
- **CMSEdit**
- **CMSFormControls** (all form controls selected in the export process are exported with any site)
- **CMSGlobalFiles** (exports with any website, needs to be created manually; the folder is exported if the 'Export global folders' option is checked during the export process)
- **CMSImportFiles**
- **CMSInlineControls** (all inline controls selected in the export process are exported with any site)
- **CMSInstall**
- **CMSMasterPages**
- **CMSMessages**
- **CMSModules**
  - Forums\Controls\Layouts\**Custom** (forum custom layouts; exported with any website, needs to be created manually; the folder is exported only if the 'Copy forum custom layouts folder' option is checked during the export process)
  - **<module name>** (folders of custom modules)
- **CMSPages**
- **CMSResources**
- **CMSScripts** (**CMSScripts\Custom** is a custom file folder and is part of the export package - the folder contains JavaScript files)
- **CMSSiteManager**
- **CMSSiteUtils**
- **CMSTemplates** (all files for selected ASPX page templates are exported with any site, page templates in other folders are exported as well if they are assigned/used on the given site; the whole folder is exported if the 'Export ASPX templates folder' option is checked in the export process)
- **CMSVirtualFiles** (stores virtual objects when using [deployment mode](#))
- **CMSWebParts** (all web parts selected in the export process are exported with any site; if the web part uses additional files, they must be placed in a folder in format **<webpartCodeName>\_files**; files that are not registered as web parts are not included in the export package)
- **Controllers**
- **Models**
- **Views**
- **<site code name>** (exports with a given site, needs to be created manually or may be created automatically when storing files on the disk; the folder is exported if the 'Export site folders' option is checked in the export process)
  - Files (the default folder for storing files if the system is configured for saving files on the disk)

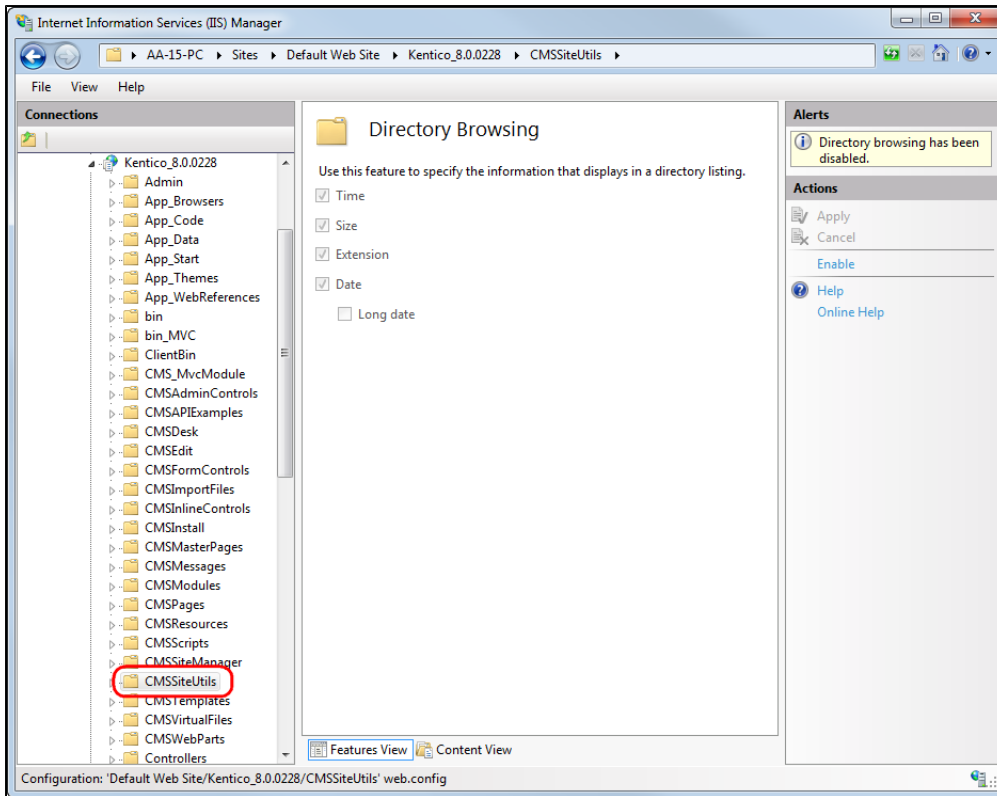
## Export/import package directory browsing

It is highly recommended to disable **Directory Browsing** in IIS for websites on live servers, at least for the **CMSSiteUtils** directory. If enabled, sensitive data from site export/import packages, such as user credentials, can be accessed directly from the browser.

Directory Browsing is disabled by default.



**Note:** To protect the CMSSiteUtils folder from enumeration attacks, disabling directory browsing is not enough. You also have to forbid access to the folder in the web.config file. See [Enumeration](#) for more security information.



## Excluding files and folders from export

You can exclude files and folders that are exported into the **Files** folder when the system creates **export** packages. Add the following keys into the *AppSettings* section of your project's *web.config* file:

- **Excluding folders from export:**

```
<add key="CMSExportExcludedFolders" value="test*;cms*" />
```

- **Excluding files from export:**

```
<add key="CMSExportExcludedFiles" value="test*;cms*" />
```

The values of the keys define names of files/folders the system excludes from the export. You can enter multiple values separated by semicolons.

You can also use the standard file system mask with the \* wildcard, which represents any number of characters in the name. No other file system mask wildcards are supported.

If the keys are not entered in the *web.config* file, files with the **.scc** extension and folders with the **.svn** extension are excluded by default. However, if the keys are used, only the specified files and folders are excluded.

## Exporting sites

The system allows you to export entire websites including their settings and related objects (such as page types, workflows, web parts, page templates, etc.) into a single file archive. You can then import the archive on any Kentico instance.

### Exporting a site

1. Open the **Sites** application.
2. Click **Export site** (📁) for the site you want to export. The Export wizard starts.
3. Fill in the name of the export package and choose the object pre-selection type:

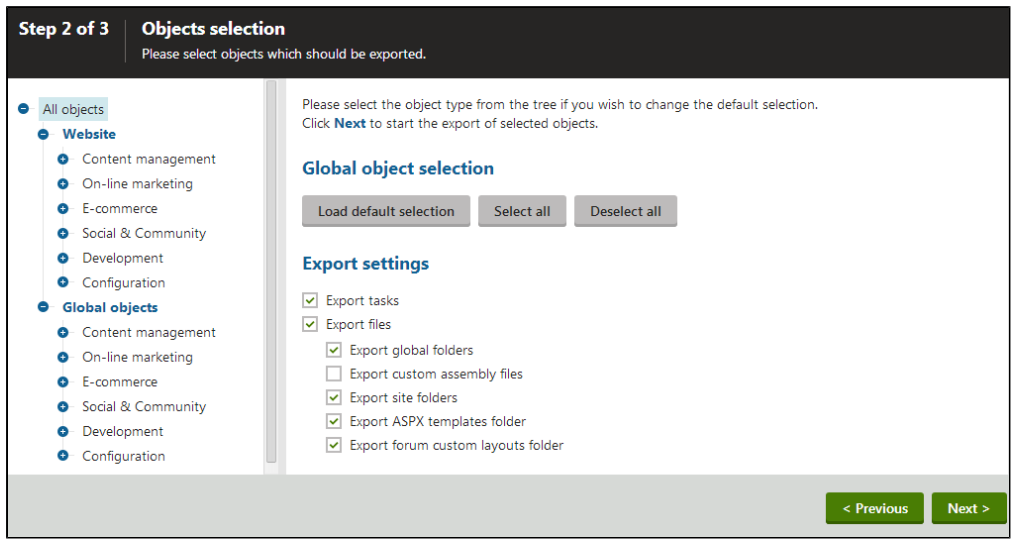
Option	Description
File name	Sets the name of the export package. The wizard creates a zip file containing the export data in the <i>~/CMSSiteUtils/Export</i> folder.

Preselect all objects	Preselects all objects related to the site.
Preselect objects changed after specific date	Preselects all objects changed after the specified date.
Use previous export settings	Preselects objects based on the settings of a previous export.

Select an option and click **Next**.

- Select which objects you want to export. The tree separates objects into categories according to the related functionality. These reflect the product categories to which the objects are related. The **Website** category contains objects related to the selected website. The **Global objects** category contains global objects that can be used by all sites. Selecting a category opens a set of check boxes in the right part of the screen, allowing you to select which objects will be exported.

 See the [Export - Object selection settings](#) section for details.



- After making all required selections, click **Next** to proceed. A log appears, showing you the export progress. You can abort the export by clicking **Cancel** at any time. When the export finishes, a message appears at the top of the log, showing the full path to the exported file.
- Click **Finish**.

The system returns you to the **Sites** application. You can now copy the exported package into the `~/CMSiteUtils/Import` folder of the target installation of Kentico, and use the **Import site or objects** wizard described in [Importing a site or objects](#).

## Export - Object selection settings

The following options are available in the **Object selection** step of the Export wizard.

Select the root of the tree (*All objects*) to access the following general export settings:

Global selection	
Load default selection	Preselects all objects based on your selection in Step 1 of the export wizard.
Select all objects	Selects all available objects.
Deselect all objects	Unselects all objects.
Export settings	
Export tasks	If enabled, delete tasks (incremental deployment) will be included in the package.
Export files	Some database objects are linked with physical files stored on the file system inside the web project. If you check this option, such files will be exported along with the corresponding database objects.

Export global folders	<p>If enabled, all files under the following folders will be added to the export package:</p> <ul style="list-style-type: none"> <li>~/App_Code/Global/</li> <li>~/CMSGlobalFiles/</li> <li>~/CMSScripts/Custom/</li> <li>~/App_Code/Controllers/</li> <li>~/Controllers/Global/</li> <li>~/Views/Global/</li> <li>~/Models/Global/</li> </ul>
Export custom assembly files	<p>If enabled, bound custom assembly files will be exported together with notification gateways, payment options, integration connectors, scheduled tasks and smart search indexes. Custom assemblies are those whose names do not begin with the CMS. prefix.</p>
Export site folders	<p>This option is only available when exporting websites, not just separate objects. If enabled, all files under the following site-related folders will be added to the export package:</p> <ul style="list-style-type: none"> <li>~/App_Code/&lt;site code name&gt;/</li> <li>~/App_Data/&lt;site code name&gt;/</li> <li>~/&lt;site code name&gt;/</li> <li>~/Controllers/&lt;site code name&gt;/</li> <li>~/Views/&lt;site code name&gt;/</li> <li>~/Models/&lt;site code name&gt;/</li> </ul>
Export ASPX templates folder	<p>If enabled, the folder with ASPX page templates will be exported:</p> <ul style="list-style-type: none"> <li>~/CMSTemplates/</li> </ul>
Export forum custom layouts folder	<p>If enabled, the folder with custom forum layouts will be exported:</p> <ul style="list-style-type: none"> <li>~/CMSModules/Forums/Controls/Layouts/Custom/</li> </ul>

The following categories contain extra options to be set:

<b>Custom tables</b>	
Export custom table data	If checked, custom table records (the actual data stored in the tables) will be exported along with the selected custom tables.
<b>Pages</b>	
Export pages	If checked, pages will be exported
Export page histories	If checked, histories of all exported pages will be exported.
Export page relationships	If checked, relationships of all exported pages will be exported.
Export page level permissions	If checked, page security settings made in the <b>Pages</b> application will be exported.
Export blog comments	If checked, blog comments will be exported.
Export event attendees	If checked, event attendees will be exported for all exported events.
<b>Forms</b>	
Export forms data	If checked, stored forms' data will be exported together with the exported forms.
Export physical files	If checked, physical files saved within form records (if there are some) will also be imported.
<b>Forums</b>	
Export forum posts	If checked, forum posts will be exported together with the exported forums.
<b>Message boards</b>	
Export board messages	If checked, board messages will be exported together with particular message boards.
<b>Media libraries</b>	
Export media files	If checked, media files stored in the database will be exported.

Export physical files

If checked, physical media files stored in the file system will be exported. This option is not selected by default as it may cause the package size grow extremely large; instead, it is recommended to export these files manually.



If you are using [content staging](#), and have the **Log export tasks** setting enabled, a list of object deletion tasks may also be displayed at the bottom of the list. If you leave the boxes checked, the objects will be deleted after you import the package on the target server.

## Exporting objects

In addition to exporting entire websites, you can create export packages containing only selected objects (web parts, page types, page templates, etc.).

### Exporting selected objects

1. Open the **Sites** application.
2. Click **Export** in the page header. The Export wizard starts.
3. Fill in the name of the export package and choose the object pre-selection type:

Option	Description
File name	Sets the name of the export package. The wizard creates a zip file containing the export data in the <code>~/CMSSiteUtils/Export</code> folder.
Preselect all objects	Preselects all objects related to the site.
Preselect objects changed after specific date	Preselects all objects changed after the specified date.
Use previous export settings	Preselects objects based on the settings of a previous export.

Select an option and click **Next**.

4. Select which objects you want to export. The tree separates objects into categories according to the related functionality. Selecting a category opens a set of check boxes in the right part of the screen, allowing you to select which objects will be exported.



See the [Export - Object selection settings](#) section for details.

**Step 2 of 3** **Objects selection**  
Please select objects which should be exported.

Please select the object type from the tree if you wish to change the default selection. Click **Next** to start the export of selected objects.

**Global object selection**

Load default selection Select all Deselect all

**Export settings**

- Export tasks
- Export files
  - Export global folders
  - Export custom assembly files
  - Export ASPX templates folder
  - Export forum custom layouts folder

< Previous Next >

5. After making all required selections, click **Next** to proceed. A log appears, showing you the export progress. You can abort the export by clicking **Cancel** at any time. When the export finishes, a message appears at the top of the log, showing the full path to the exported file.
6. Click **Finish**.

The system returns you to the **Sites** application. You can now copy the exported package into the `~/CMSSiteUtils/Import` folder of the target installation of Kentico, and use the **Import site or objects** wizard described in [Importing a site or objects](#).

### Export - Object selection settings

The following options are available in the **Object selection** step of the Export wizard.



Select the root of the tree (*All objects*) to access the following general export settings:

<b>Global selection</b>	
Load default selection	Preselects all objects based on your choices in Step 1 of the import wizard.
Select all objects	Selects all available objects.
Deselect all objects	Unselects all objects.
<b>Export settings</b>	
Export tasks	If enabled, delete tasks (incremental deployment) will be included in the package.
Export files	Some database objects are linked with physical files stored on the file system inside the web project. If you check this option, such files will be exported along with the corresponding database objects.
Export global folders	If enabled, all files under the following folders will be added to the export package: <ul style="list-style-type: none"> <li>• ~/App_Code/Global/</li> <li>• ~/CMSGlobalFiles/</li> <li>• ~/CMSScripts/Custom/</li> <li>• ~/App_Code/Controllers/</li> <li>• ~/Controllers/Global/</li> <li>• ~/Views/Global/</li> <li>• ~/Models/Global/</li> </ul>
Export custom assembly files	If enabled, bound custom assembly files will be exported together with notification gateways, payment options, integration connectors, scheduled tasks and smart search indexes. Custom assemblies are those whose names do not begin with the <i>CMS.</i> prefix.
Export ASPX templates folder	If enabled, the folder with ASPX page templates will be exported: <ul style="list-style-type: none"> <li>• ~/CMSTemplates/</li> </ul>
Export forum custom layouts folder	If enabled, the folder with custom forum layouts will be exported: <ul style="list-style-type: none"> <li>• ~/CMSModules/Forums/Controls/Layouts/Custom/</li> </ul>

The following categories contain extra options to be set:

<b>Custom tables</b>	
Export custom table data	If checked, custom table records (the actual data stored in the tables) will be exported along with the selected custom tables.
<b>Pages</b>	
Export pages	If checked, pages will be exported
Export page histories	If checked, histories of all exported pages will be exported.
Export page relationships	If checked, relationships of all exported pages will be exported.
Export page level permissions	If checked, page security settings made in the <b>Pages</b> application will be exported.
Export blog comments	If checked, blog comments will be exported.
Export event attendees	If checked, event attendees will be exported for all exported events.
<b>Forms</b>	
Export forms data	If checked, stored forms' data will be exported together with the exported forms.
Export physical files	If checked, physical files saved within form records (if there are some) will also be imported.
<b>Forums</b>	

Export forum posts	If checked, forum posts will be exported together with the exported forums.
<b>Message boards</b>	
Export board messages	If checked, board messages will be exported together with particular message boards.
<b>Media libraries</b>	
Export media files	If checked, media files stored in the database will be exported.
Export physical files	If checked, physical media files stored in the file system will be exported. This option is not selected by default as it may cause the package size grow extremely large; instead, it is recommended to export these files manually.



If you are using [content staging](#), and have the **Log export tasks** setting enabled, a list of object deletion tasks may also be displayed at the bottom of the list. If you leave the boxes checked, the objects will be deleted after you import the package on the target server.

## Exporting single objects

You can export some Kentico objects separately as single object packages. For example, single object export is supported for:

- [CSS stylesheets](#)
- [Page types](#)
- [E-mail templates](#)
- [Form controls](#)
- [Page layouts](#)
- [Page templates](#)
- [Web part containers](#)
- [Web parts](#).

To export an object, navigate to the appropriate application in the Kentico administration interface.

For objects in lists, click **Other actions** (...) and select **Export** in the menu.

Actions	Display name ▲	Code name
...	Article	CMS.Article
	Clone	CMS.Blog
	Destroy	CMS.BlogMonth
	Export	CMS.BlogPost
	Backup	CMS.CellPhone
	Restore	Chat.Transformations
...	Community - Transformations	Community.Transformations

For objects organized in tree menus, you can select the object and click **Export object** () above the tree.

The **Export object** dialog opens.

1. Type a name for the export file (a default name will be pre-entered).
2. Click **Export**.

When the object export is successfully finished, the dialog shows the path to the exported file. You can click **Download package** to save the file into a different location.

## Importing a site or objects

After you export a site or objects using the [Export site wizard](#) or [Export objects wizard](#), you can import the content using the **Import wizard**. Before you start the wizard, you need to copy your export packages into the application's `~/CMSiteUtils/Import` folder.

## Importing a site

1. Open the **Sites** application.
2. Click **Import site or objects**. The Import wizard starts.
3. Select the package that you want to import.
  - The list contains all packages stored in the `~/CMSiteUtils/Import` folder.
  - You can upload additional packages to the folder by clicking **Upload package**.
4. Choose one of the following options once you select a package:

Option	Description
Preselect all items	Preselects all objects in the package.
Preselect only new items	Preselects only objects that do not already exist in the system.

5. Click **Next**.
6. When importing a site package, select one of the following options. The wizard skips this step if you are importing a package containing only global objects.

Option	Description
Import a new site	A new site will be created based on the contents of the package. You need to enter the site's display name, code name and domain name.
Import objects into an existing site	The wizard imports the content of the package into the selected site.

7. Click **Next**.
8. Select which objects from the package you want to import. Choose the categories displayed in the tree on the left side of the wizard.



See the [Import - Object selection settings](#) section for details about the available options.

Objects that already exist in the target system are marked with an asterisk (\*). If you import an existing object, the new version overwrites the current one.

9. Click **Next** to execute the import process. An import log appears, showing the progress of the import (you can abort the import process at any time by clicking **Cancel**).
10. Click **Finish**.

The system returns you to the **Sites** application.



### Importing packages from older Kentico versions

Packages from different versions of Kentico have a different structure. When importing packages from an older version of Kentico to a newer one, the system automatically converts the package to the newer format.

- If possible, avoid importing **Form control** and **Web part** objects from older packages, which may cause incompatibility problems and errors.
- When importing sites from older versions, you may need to assign the system's modules to the site after the import:
  1. Open the **Sites** application.
  2. Edit the imported site.
  3. Add all required modules on the **Assigned objects -> Modules** tab.



When importing to instances installed in the web application format, you may need to perform additional steps to complete the import. See [Importing to web application projects](#).

## Conflicts of running sites

If the imported site uses the same domain name or alias as one of the websites already running in your system, the import displays a warning at the end and the imported site does not start.

In such cases, you need to:

1. Open the **Sites** application.
2. Edit ( ) the imported site and change the conflicting site's **Site domain name** or **Domain alias**.
3. Return to the list of sites and click **Start site** ( ) next to the imported site.



### Application restart

If you get the following error message at the end of the import process:

*"Application has been restarted and the logging of the import process has been terminated. Finish the import process manually."*

you need to finish the import process manually:

1. Open the import package and extract all content of the `<package>\Data\Files` folder.
2. Remove the `.export` extension from the names of all files in all extracted sub-folders.
3. Rename all folders named `##SITENAME##` to the code name of the target website and copy them to the root of the web project.
4. Copy all contents of the `cms_attachments` folder (if it is present among the extracted folders) to the location in the target project where page attachments are stored (as configured in [Settings -> System -> Files](#)). Please note that this is applicable only when the system is configured to store page attachments in the file system (not in the database).
5. Copy the content of all folders named as an object type (e.g. `cms_avatar`, `cms_documenttype`, `forums_forum`, etc.) to the root of your web project.

## Re-signing imported macro expressions

The system uses signatures to ensure the security of [macro expressions](#). The signatures are only valid in the environment of the application where the macros were originally saved.


To ensure that macros inside the data of imported sites or objects work correctly, you need to re-sign the macros using the **System -> Macros -> Signatures** interface. See [Working with macro signatures](#) for additional information.

## Import - Object selection settings

The following options are available in the **Object selection** step of the Import wizard.


Select the root of the tree (*All objects*) to access the following import settings:

Global selection	
Load default selection	Preselects all objects based on your choices in Step 1 of the import wizard.
Select all	Selects all objects in the import package.
Select only new	Selects only those objects that do not exist in the target database (existing objects are unselected).
Deselect all	Clears selection for all objects in the import package.
Import settings	
Update site definition	Displayed only when importing to an existing site. If enabled, all settings stored as part of the site object will be updated with those contained in the package. These settings are stored in the <code>Site\cms_site.xml</code> file inside the export package.

Assign all objects to the imported site (recommended)	If enabled, all imported site-related objects will be assigned to the imported site.
Run the site after import	If checked, the system attempts to start the new site immediately after the import is finished.
Delete incomplete site when import fails	If checked, the system deletes the site if any part of the import process fails.
Do not import objects where parent object is missing	If enabled, the import process skips any child objects whose parent objects is not present in the target system.
Import tasks (recommend)	If enabled, <a href="#">synchronization tasks</a> included in the package will be performed.
Log staging synchronization tasks	If enabled, the system logs staging tasks reflecting all changes made by the import. Check this option to synchronize the imported data to other servers connected through <a href="#">Content staging</a> .
Log integration tasks	If enabled, the system logs outgoing integration tasks for all changes made by the import. Check this option if you want to transfer the imported data to a system connected via the <a href="#">System integration bus</a> .
Import files (recommended)	Some database objects are linked with physical files stored on the file system inside the web project. If you check this option, the import process creates all files that are included in the import package.
Import code files	Indicates if the import includes code files that require compilation, i.e. files with the following extensions: <i>cs</i> , <i>vb</i> , <i>aspx</i> , <i>ascx</i> You cannot import code files if your application is <a href="#">precompiled</a> .
Import custom assembly files	If enabled, the import process includes custom assembly files bound to notification gateways, payment options, integration connectors, scheduled tasks and smart search indexes included in the import package. Custom assemblies are those whose names do not begin with the <i>CMS</i> . prefix. You cannot import assembly files if your application is <a href="#">precompiled</a> .
Import global folders	If checked, the import process includes global files originally exported from the following folders: <ul style="list-style-type: none"> <li>• ~/App_Code/Global/</li> <li>• ~/CMSGlobalFiles/</li> <li>• ~/CMSScripts/Custom/</li> <li>• ~/App_Code/Controllers/</li> <li>• ~/Controllers/Global/</li> <li>• ~/Views/Global/</li> <li>• ~/Models/Global/</li> </ul>
Import site folders	If checked, the import process includes site-related files originally exported from the following folders: <ul style="list-style-type: none"> <li>• ~/App_Code/&lt;site code name&gt;/</li> <li>• ~/App_Data/&lt;site code name&gt;/</li> <li>• ~/&lt;site code name&gt;/</li> <li>• ~/Controllers/&lt;site code name&gt;/</li> <li>• ~/Views/&lt;site code name&gt;/</li> <li>• ~/Models/&lt;site code name&gt;/</li> </ul>
Overwrite system queries	Available only when importing from an older version of Kentico. If checked, the system imports all queries from the package and overwrites the current ones. <div style="border: 1px solid red; padding: 5px; margin-top: 10px;">  If the <b>package contains custom queries</b> that you added to the system, it is necessary to have this option enabled. </div>

The following object categories contain extra options that you can set:

#### Automation processes

Import all site specific triggers	If checked, all site-specific triggers will be imported, even if the respective processes are not selected. This check-box is only displayed when the import package contains an exported website, not only separate objects. If you are importing to an existing site, this option is deselected by default.
<b>Custom tables</b>	
Import custom table data	If checked, custom table records (the actual data stored in the tables) included in the package will be imported together with the respective custom tables. If you are importing to an existing site, this option is disabled by default.
<b>Pages</b>	
Import new pages	If enabled, pages included in the import package will be imported. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> When importing into an existing site, only new pages can be imported. Pages that are already present on the target server will not be overwritten.</div>
Import page histories	If enabled, page histories (i.e. previous versions of pages) will be imported.
Import page relationships	If enabled, page relationships will be imported.
Import page-level permissions	If enabled, page security settings made in the <b>Pages</b> application on the <b>Properties -&gt; Security</b> tab will be imported.
Import blog comments	If enabled, blog comments will be imported together with blog post pages.
Import event attendees	If checked, event attendees will be imported together with all imported events.
Import all user personalization	If checked, user personalization of all users will be imported, even for users who are not selected to be imported. If you are importing to an existing site, this option is not selected by default.
<b>Forms</b>	
Import form data	If enabled, forms data included in the package will be imported together with the forms.
Import physical files data	If checked, physical files saved within form records (if there are some) will also be imported.
<b>Forums</b>	
Import forum posts	If enabled, forum posts included in the package will be imported together with the forums.
<b>Groups</b>	
Import all group memberships	If enabled, user memberships will be imported together with the selected groups.
<b>Message boards</b>	
Import board messages	If enabled, board messages included in the package will be imported together with the message boards.
<b>Media libraries</b>	
Import media files	If enabled, media files (stored in database) included in the package will be imported together with the media libraries.
Import physical files	If enabled, physical files (stored in the file system) included in the package will be imported together with the media libraries.
<b>Page templates</b>	
Import all site specific page template scopes	If checked, site-specific page template scopes will be imported for all page templates, even if a page template itself is not selected to be imported. This check-box is only displayed with page templates in the <b>Global</b> section of the tree and only when the import package contains an exported website, not only separate objects. If you are importing to an existing site, this option is de-selected by default.

Import web part and zone variants with the selected page templates	If checked, web part and zone variants (from <a href="#">MVT testing</a> or <a href="#">Content personalization</a> ) included in the package will be imported together with the selected page templates.
<b>Users</b>	
Import user dashboards	Indicates if the system imports the content of personalized <a href="#">widget dashboards</a> together with imported users. If you are importing to an existing site, this option is de-selected by default so that existing dashboards are not overwritten.
Import all user site specific dashboards	If checked, the content of all personalized <a href="#">widget dashboards</a> is imported, even for users who are not selected in the import options. This option appears only if the import package contains an exported site, not just separate objects. If you are importing to an existing site, this option is de-selected by default so that existing dashboards are not overwritten.
<b>Workflows</b>	
Import all workflow scopes	If checked, all workflow scopes will be imported, even if the respective workflows are not selected. Import web part and zone variants with the selected page templates. This check-box is only displayed when the import package contains an exported website, not only separate objects. If you are importing to an existing site, this option is not selected by default.



If you are using [content staging](#), and have the **Log export tasks** setting enabled in **Settings -> Versioning & Synchronization -> Staging**, a list of object deletion tasks may also be displayed at the bottom of the list. If you leave the boxes checked, the objects will be deleted after you import the package on the target server.

overwriting your current objects of these types, as it may cause incompatibility problems.

## Importing to web application projects

Due to the [differences between website and web application](#) projects, the system cannot import from one to another automatically. After completing the standard import procedure, manually perform the following steps to ensure that your web application runs correctly:

1. The import places all physical files from the imported package into the `~/App_Data/CMSTemp/ImportExport/Files` folder. Copy the content of the folder into the **CMS** folder of your web application project.



We strongly recommend creating a backup of your project before you overwrite any files.

2. Open the solution file (*WebApp.sln*) in Visual Studio from the web project directory.
3. Include the files that did not exist in the project before the import:
  - a. Click **Show all files** at the top of the Solution Explorer.
  - b. Locate the newly added files.
  - c. Select the files you wish to include one-by-one while holding the **Ctrl** key.
  - d. Right-click one of the files and select **Include in Project**.
4. Right-click the project node (CMSApp) in the Solution Explorer, and select **Convert To Web Application**.
5. Rebuild the solution.



If you are importing a package from another web application project, you do not need to convert the files (step 4), but it is still necessary to include the imported files in the project.

## Importing into precompiled projects

If you are running a [precompiled website](#), you cannot import:

- Physical files that require compilation (such as *cs*, *vb*, *aspx*, *ascx*)
- The **App\_Code** folder
- Virtual objects that contain ASP.NET markup:
  - [Page layouts](#) (ASCX type)
  - [Transformations](#) (ASCX type)
  - [Web part layouts](#)
  - [Form layouts](#) (ASCX type)

To add files or objects that require compilation, you need to perform the import on the original project and run the precompilation again.

## Publishing web application projects from Visual Studio

If you wish to use the Visual Studio Publish function (MSDeploy), we recommend installing Kentico projects with the *web site* configuration.

The Kentico solution consists of multiple web application projects (**CMSApp**, **CMSApp\_AppCode**, **CMSApp\_MVC**). To use the **Publish** function of Visual Studio for Kentico web application projects, you need to publish all of the projects to the same target location to ensure that the solution works correctly.



### Note

After deploying the first project, make sure that your publish profile has the **Delete all existing files prior to publish / Remove additional files at destination** option disabled. You can configure the option in the **Publish Web** dialog in **Settings -> File Publish Options**.