

When using [continuous integration](#), you can get changes from other developers or your source control system by loading XML files into your *CIRepository* folder. You then need to transfer the corresponding object changes to the database of your Kentico instance. Kentico provides a command line utility that deserializes the files and saves the object data back into the database.

#### **Warning**

The restore operation **deletes** any [supported objects](#) in the database that do not exist as files in the continuous integration repository. Before you run the restore process, make sure the *CIRepository* folder contains the required state of your object data.

To avoid loss of local data, you need to maintain the file system repository as a full image of your database – serialize all objects to the file system at the start of development and leave continuous integration enabled. You then need to merge the file content of the *CIRepository* folder when loading data from another team member or the central source control.

Navigate to the **CMS\bin** folder of your Kentico project and run **ContinuousIntegration.exe** from the command line with the **-r** parameter.

```
ContinuousIntegration.exe -r
```

The utility deserializes the objects stored in the project's *CIRepository* folder and creates, overwrites or removes corresponding data in the database (specified by the connection string in the given project's web.config file).

To ensure that the restore process works correctly, you need to **stop your Kentico application before running the restore process**. Otherwise you may encounter the following problems:

- Deadlocks or data inconsistencies if the system attempts to write to the *CIRepository* folder while data is being restored from the files
- Outdated content in the application's cache if you restore without restarting (can cause inconsistencies in the Kentico administration interface or the website's content)

**Note:** If you are using a web application project and always rebuild your solution after loading files from the source control, the application stops and starts automatically.

## Rebuilding smart search indexes

Restoring data from the *CIRepository* folder may create new [smart search indexes](#) or cause existing ones to become outdated.

If you wish to automatically keep your site's search functionality up-to-date after continuous integration restore operations, run **ContinuousIntegration.exe** from the command line with the additional **-s** parameter.

```
ContinuousIntegration.exe -r -s
```

When restoring data with the **-s** parameter, the system automatically logs rebuild tasks for:

- Any new search indexes
- All indexes whose indexing configuration was modified by the changes

For example, if a restore operation updates the field search settings of a page type, the restore utility logs rebuild tasks for all page search indexes that include the given page type. The index rebuilds are processed when the application starts up after the restore operation.



**Note:** We do not recommend using the **-s** restore parameter if you have large search indexes with very long rebuild times. In these cases, you can rebuild your indexes manually as required.

## Scripting the restore operation

We recommend setting up a process that runs the restore operation, stops/starts your application and performs any other required actions. For example, you can prepare a [PowerShell](#) script.

### Basic PowerShell example

```
# Parameter that specifies the path of your Kentico web project folder (i.e. the CMS subfolder)
param (
    [Parameter(Mandatory=$true)]
    [string]$Path
)

# Creates an 'App_Offline.htm' file to stop the website
"<html><head></head><body>Continuous integration restore in progress...</body></html>"
> "$Path\App_Offline.htm"

# Runs the continuous integration restore utility
& "$Path\bin\ContinuousIntegration.exe" -r

# Removes the 'App_Offline.htm' file to bring the site back online
Remove-Item "$Path\App_Offline.htm"
```

The script above is only a basic example that demonstrates the general concept. We recommend that you adjust the script to perform all functions required by your environment. The attached [Restore-CI.ps1](#) file contains a more complete script that also stops and starts Kentico external services.



### Custom restore solutions

Alternatively, you can use the Kentico API to implement a custom solution for restoring objects from the file system repository. See [Restoring objects from the file system](#).