The **Kentico Code upgrade tool** provides a way to analyze Kentico 10 projects and helps convert custom code to version 11. You can download the tool from the API Changes page on the DevNet portal.

The tool has the following functionality:

- Detects custom code that is no longer valid in version 11
- Provides recommendations for each occurrence of invalid code
- Automatic conversion of the majority of code to the Kentico 11 API

Prerequisites

The machine where you plan to run the code upgrade tool must fulfill the following requirements:

- Microsoft .NET Framework 4.5.1 or newer
- Development and compilation tools:
  - Visual Studio 2015
    – OR –
  - Visual Studio 2017 with Build Tools for Visual Studio 2017 (including *Web development build tools*)

The Kentico 10 solution that you wish to analyze must compile without errors.

To allow the tool to check the code of Kentico virtual objects, such as transformations and page layouts, you need to enable Deployment mode for your Kentico 10 instance.

Analyzing Kentico 10 projects

Basic detection

Open the *Tool_VS2015* or *Tool_VS2017* folder in the code upgrade tool package, based on the Visual Studio version that you use on your machine.

Run **CodeUpgrade.exe** from the **command line**, with the path to your project's solution file as the parameter (*WebSite.sln* or *WebApp.sln*).

For example:

```
CodeUpgrade.exe C:\inetpub\wwwroot\Kentico10\WebSite.sln
```

The tool generates a **csv** file containing a list of custom code occurrences in your project that are no longer valid in Kentico 11. The information will help you update your custom code after you perform the upgrade.

Each listed occurrence in the csv file has one of the following statuses:

- **Detected** – an occurrence of invalid code that the tool cannot resolve automatically. See the *Instructions* value of the occurrence for more information.
- **CanBeResolved** – a detected code occurrence that the tool can resolve automatically if you run it with the *-resolve* parameter (see the *Automatic conversion* section).
- **Resolved** – invalid code that the tool automatically resolved in the specified output folder. Only used when running the tool with the *-resolve* parameter.

Automatic conversion

You can also use the tool to provide automatic resolutions for most types of code expressions. Add the **-resolve** parameter when running *CodeUpgrade.exe*.

For example:

```
CodeUpgrade.exe -resolve C:\inetpub\wwwroot\Kentico10\WebSite.sln
```

If the tool detects code issues that can be automatically resolved, it creates copies of the given files under the **Converted** folder in the tool directory. All resolvable issues in the files are updated to be compatible with the Kentico 11 API.

Optional parameters

You can also add the following parameters when running the tool:

- **-vstoolspath=<path>** – only available when using the Visual Studio 2017 version of the tool. Sets the path to the Visual Studio build tools required to analyze the solution. You need to set the path if your build tools are installed in a different directory than the default (*C:\Program Files (x86)\Microsoft Visual Studio\2017\BuildTools\MSBuild\Microsoft\VisualStudio\v15.0*).
- **-output=results.csv** – sets the name of the file containing the tool's results.
- **-verbose=0** – determines how detailed the tool's output is in the console (enter values from 0 to 3). Does not affect the output file.

For more details about the available parameters, run the tool without any parameters.

Limiting the scope of the analysis

For customized Kentico files, the tool reports all detected issues, including the default Kentico 10 code. You can avoid reporting of these false positives by enclosing your custom code into **#region** directives with a specific name. For ASPX markup, use the following syntax:

```
<%-- #region Name --%> ... <%-- #endregion --%>
```

To analyze only the code placed inside the regions, run the tool with an additional parameter: **-customregion=<regionName>**

Implementing the API changes

Use the output of the tool to update your custom code AFTER you upgrade the project to Kentico 11.

The output file lists all occurrences of invalid custom code, including:

- The path of the affected file and line number
- The exact source code of the expression
- A set of instructions on how to resolve the issue

If you generated automatic conversions for your code, use the following approaches to insert the new code into your upgraded project:

- **Customized Kentico files** – use a code comparison tool and merge the Kentico 11 files with the matching files in the *Converted* output folder.
- **Fully custom files** – directly replace the files using the converted files created by the tool.

Tip: To update the code of virtual objects, enable Deployment mode. You can then fix the code in the corresponding physical files, which match the output of the code upgrade tool. Disable deployment mode to return the updated virtual objects back into the database.