

Table Of Contents

- AxCMS.net 5.x
 - 5.5.1
 - 5.5.0
 - Upgrade to VS 2005
 - 5.2.8
 - 5.2.7
 - 5.2.5
 - 5.2.3
 - 5.2.2
 - 5.2.1
 - Init_Sample

AxCMS.net upgrade to 5.x version

Upgrade to AxCMS.net 5.5.1

New Language-Pack

Please download the latest version of the AxCMS.net Language Pack. It is necessary even if you do not plan to use other languages as german.

Please also change your web.config. The resource manager "CMS" should get the language "ax" as a default language. Put is also into the list of supported languages. The first language in the list is the assumed target language, if no language is specified by user explicitly. Axinom recommends this ResourceManager configuration:

```
<ResourceManager name="CMS"> <DefaultLanguage>ax</DefaultLanguage>
<Languages>en , ax , de</Languages>
<InsertLanguage>>false</InsertLanguage>
<IsEditable>>false</IsEditable>
<Notification>>false</Notification>
<CheckDefaultLanguage>>false</CheckDefaultLanguage>
<Table>CMS</Table>
</ResourceManager>
```

AxUser refactoring

If you are developer of a template project you may have to change your code because of a breaking change concerning AxUserDataAgent.

This class has been splitted into two: the new AxUserDataAgent (capable to manage user categories, priviledges and rights) and the AxUserAdapter

responsible for persistence and authentication. The AxUserAdapter uses the AxCMS.net Persistence Framework.

New configuration section for WebQueryService

Please add the following custom section registration into configSections of the web.config of your management system:

```
<section
name="WebQueryService"
type="Axinom.AECMS.Gateway.GatewayConfiguration, Axinom.AECMS.Interfaces"
/>
```

and the section itself:

```
<WebQueryService type="WebService">
<Url>http://localhost/AxCMSPublishService/WebQueryService.asmx</Url>
<Timeout>30</Timeout> <!-- seconds -->
</WebQueryService>
```

Known Issues

If you [extend the live AxUser](#) you cannot use fields that doesn't allow nulls. Either declare the field as NULL or declare it as NOT NULL and provide a DEFAULT value. This bug is fixed in 6.0.0.

Upgrading to AxCMS.net 5.5.0

Upgrading to Visual Studio 2005

AxCMS.net 5.5.0 is the first AxCMS-version designed to run with .NET Framework 2.0 and Visual Studio 2005. Before you upgrade to AxCMS.net 5.5.0, please upgrade your .NET-environment. Check **Prerequisites** from the [download-page](#). Read more about upgrading at "[Upgrading to Visual Studio 2005](#)". If you already have a project running under previous versions of AxCMS.net, carefully follow the instructions about converting your project.

Scheduling Reports

With AxCMS.net 5.5.0 you can schedule the CMS-reports. Add this key to your web.config:

```
<!-- Time to execute EverydayAtSpecifiedTime watcher -->
<add key="ExecuteAt" value="13:40" />
```

If you don't add this key, reports are sent as usual - at midnight every day.

enableEventValidation

The new ASP.NET offers one more security feature: it verifies that arguments to postback or callback events originate from the server control that originally rendered them. If the verification fails, "Invalid postback or callback argument."-exception is thrown.

Microsoft recommends using the ClientScriptManager.RegisterForEventValidation method in order to register the postback or callback data for validation.

If you get this exception but have no control over the code causing it, you can turn off event validation by putting this line into web.config:

```
<pages validateRequest="false" enableEventValidation="false"/>
```

Ip Authentication User Interface

If you have problems with accessing UI for IP Authentication in MS, then have a look in the web.config of MS and remove (if existing)

following keys in section "IpRuleService":

```
<Username>Username</Username>
<Password>Password</Password>
```

Note: If you don't have NUnit 2.2 installed and you are using Sample version 2.0, you'll need the file nunitframework.dll for Init_Sample to work correctly. It must be located under Init_Sample folder.

You can download the necessary file [here](#).

WebResource.axd handler does not work (LinkButton is not clickable) with AuthenticationModule

If you use AuthenticationModule in the Live System, it intercepts all the requests. Requests to WebResource.axd too. This file does not exist on the hard drive, this is why AuthenticationModule returns HTTP error 404 (not found).

WebResource.axd is a new feature of ASP.NET 2.0 (read, for example, [this article](#)).

ASP.NET 2.0 sometimes uses WebResources.axd to load its JavaScript-files, for example LinkButton uses it. It means, some problems with JavaScript-enabled ASP-Controls are possible, if AuthenticationModule is used.

Workaround: Create a dummy file WebResource.axd in the root of the Live System.

The problem will be addressed in the future versions of AxCMS.net.

Upgrading to Visual Studio 2005

Visual Studio 2005 and AxCMS.net

All the versions of AxCMS.net until 5.2.X have been based on .NET Framework 1.1. They have been designed to use with Visual Studio .NET 2003. However, on December 30th, 2005 Axinom released the first AxCMS.net ([version 5.5.0](#)) targeted to .NET Framework 2.0 and Visual Studio 2005. This chapter gives you an overview of the upgrade procedure.

Upgrading development environment

Your development environment should be prepared for the upgrade the following way:

- 1) Install Microsoft Visual Studio 2005 (should be Standard Edition or higher, no Express Edition).
- 2) Download and install Visual Studio 2005 Web Application Project (available at: <http://go.microsoft.com/fwlink/?LinkId=57870>).
- 3) It is also recommended (although not compulsory) that you upgrade your SQL server. One option is to use Microsoft SQL Server Express edition (available for free at: <http://msdn.microsoft.com/vstudio/express/sql/>)

Upgrading production environment

For the production environment only the .NET Framework 2.0 is necessary.

Microsoft .NET Framework 2.0 Redistributable Package is available at:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=0856EACB-4362-4B0D-8EDD-AAB15C5E04F5&displaylang=en>

Converting your AxCMS.net-templates-project

In Visual Studio 2005 Microsoft changed the way web-projects are handled, but had to re-introduce it later in form of an add on: "VS 2005 Web Application Projects". To take advantage of this add one one has to manually convert the projects, created with VS 2003. More information available [here](#). Following are the steps:

1. Backup your current solution

In order to avoid potential problems, please backup everything related to your development: source code, databases etc. It is quite likely that nothing serious will happen, but better safe than sorry.

2. Open your VS 2003 solution and make sure it compiles clean

It can save you much time troubleshooting later.

If you use source control, please check that everything is checked in, because the VS 2005 Conversion Wizard checks out some files.

3. Modify the VS 2003 .sln/.csproj files to change the web project to be a class library

3.1 Close the solution in VS 2003

3.2 Using notepad, open the .sln solution file for the solution. The .sln file is a text-based file that contains your project wide-settings, as well as pointers to all of the projects and project-files contained within it. For example, the below snippet shows a VS 2003 solution file with a web-project:

```
Microsoft Visual Studio Solution File, Format Version 8.00
Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "AxCMSTemplates_Sample",
"http://localhost/AxCMSTemplates_Sample/AxCMSTemplates_Sample.csproj", "{345F0E9A-24F3-451C-
B8EE-6B3796797BC8}"
ProjectSection(ProjectDependencies) = postProject
EndProjectSection
EndProject
```

Manually edit the web-project definitions to remove the **http://** path location information, and instead just point to the .csproj file directly (like you would for a normal class library - using file-path information relative to the .sln file). For example, the web project definition above - "http://localhost/AxCMSTemplates_Sample/AxCMSTemplates_Sample.csproj" - would change instead to be "AxCMSTemplates_Sample\AxCMSTemplates_Sample.csproj" (note that the path had to change from "/" to "\" because it is now a

file-path).

Note: if your web-projects are already referenced as csproj-files directly, without http://, go directly to the step 5.

```
Project( "{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}" ) = "AxCMSTemplates_Sample",
"AxCMSTemplates_Sample\AxCMSTemplates_Sample.csproj", "{345F0E9A-24F3-451C-B8EE-6B3796797BC8}"
ProjectSection(ProjectDependencies) = postProject
EndProjectSection
EndProject
```

3.3 Using notepad, open and edit each of the .csproj files for the web projects in your solution. They will each contain a header that

looks like this:

```
<VisualStudioProject>
<CSHARP
ProjectType = "Web"
ProductVersion = "7.10.3077"
SchemaVersion = "2.0"
ProjectGuid = "{345F0E9A-24F3-451C-B8EE-6B3796797BC8}"
>
```

Change the "ProductType" value from "Web" to "Local", and then save and close the files:

```
<VisualStudioProject>
<CSHARP
ProjectType = "Local"
ProductVersion = "7.10.3077"
SchemaVersion = "2.0"
ProjectGuid = "{345F0E9A-24F3-451C-B8EE-6B3796797BC8}"
>
```

3.4 Lastly (and importantly), delete the **.webinfo** files associated with each project. For example, if the .csproj file's name is "AxCMSTemplates_Sample.csproj", look for and delete it's "AxCMSTemplates_Sample.csproj.webinfo" file.

4. Re-open the solution in VS 2003 and verify that the projects open and build fine

Once you've completed the above steps, your web-projects will be treated as class-libraries in VS 2003. They will have the same build semantics as before - but their tie to Frontpage Server Extensions will be gone.

5. Open your development solution with Visual Studio 2005

Visual Studio automatically fires *Visual Studio Conversion Wizard* at opening the solution. Let it convert your solution to Visual Studio 2005 – it is very intuitive to use that wizard.

In the end Visual Studio displays details of conversion (make sure that all the projects in your solution have been upgraded successfully).

Choose the "Save All" command in Visual Studio's file-menu to make sure that you save the converted solution and project files and everything you have open.

6. Convert your web projects to be Visual Studio 2005 Web Application Projects

Close the solution (File->Close Solution).

Using notepad, open up the .csproj files for your web-projects again. This project file will now be in a MSBuild based file-format that looks like this:

```
<Project DefaultTargets="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
<PropertyGroup>
<ProjectType>Local</ProjectType>
<Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
<Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
<ProductVersion>8.0.50727</ProductVersion>
<SchemaVersion>2.0</SchemaVersion>
<ProjectGuid>{22C5DF64-8114-41F3-92DF-DBC285835584}</ProjectGuid>
<OutputType>Library</OutputType>
<AppDesignerFolder>Properties</AppDesignerFolder>
<RootNamespace>WebApplication10</RootNamespace>
<AssemblyName>WebApplication10</AssemblyName>
</PropertyGroup>
```

You then need to **delete** the line with ProjectType=Local

and replace it with the exact line below:

```
<ProjectTypeGuids>{349c5851-65df-11da-9384-00065b846f21}; {fae04ec0-301f-11d3-bf4b-00c04f79efbc}</ProjectTypeGuids>
```

Save and close the web project's .csproj file and re-open the .sln solution file

7. Configure IIS to use .NET Framework 2.0 for your project

Open Internet Information Services configuration panel and do the following for each template project's virtual directory:

[Right-click on the virtual directory] --> properties --> ASP.NET --> [change the ASP.NET version from 1.1 to 2.0].

Note. Here is how to do it programmatically:

```
WindowsDirectory%\Microsoft.NET\Framework\v2.0.50727\aspnet_regiis -s w3svc/1/root/%dnnname%
```

Note. Make sure you remove WebApplication from the "templates" subfolder: at IIS Configuration Manager [right-click on "templates"] --> [Properties] --> [Virtual Directory] --> [under "Application Settings" press "Remove"].

8. Upgrade AxCMS.net

Follow the regular steps to upgrade AxCMS.net to the latest version. Detailed upgrade instructions are available [here](#).

9. Check your web-proxies

If you are using web-references to access web-services, the conversion wizard will regenerate the proxy-classes. If you made any changes to them, they are lost. You have to find your previous versions in the backup (haven't we warned you?) or in your source control system and reapply the changes.

Possible changes are:

- Custom bases class for the proxy (e.g. WebServicesClientProtocol is required for WSE)
- Custom constructor (e.g. passing Url or configuration-class as a parameter)
- Deleting of some automatically generated parameter-classes

10. Compile your solution using VS 2005

You will get many warnings, caused by the switch do .NET Framework 2.0, where many old features are marked obsolete. But you should be able to compile the solution and use it under .NET Framework 2.0 and Visual Studio 2005.

That's pretty much it. After completing those steps you are ready to develop your template project by Visual Studio 2005. Enjoy!

Upgrading to AxCMS.net 5.2.8

Scheduling Reports

With AxCMS.net 5.5.0 you can schedule the CMS-reports. Add this key to your web.config:

```
<!-- Time to execute EverydayAtSpecifiedTime watcher -->  
<add key="ExecuteAt" value="13:40" />
```

If you don't add this key, reports are sent as usual - at midnight every day.

Upgrading to AxCMS.net 5.2.7

Configuring WebDAV

AxCMS.net 5.2.7 offers advanced WebDAV-support for managing documents and categories. If you have not already done it, you can configure WebDAV-support following [the instructions](#). If you already used WebDAV in AxCMS.net 5.2.6, no changes are required, if want to use Windows Authentication. If you prefer to use CMS-Authentication, check the [WebDAV-documentation](#).

Classifier.ImFeelingLuckyNavigation(int pageID)

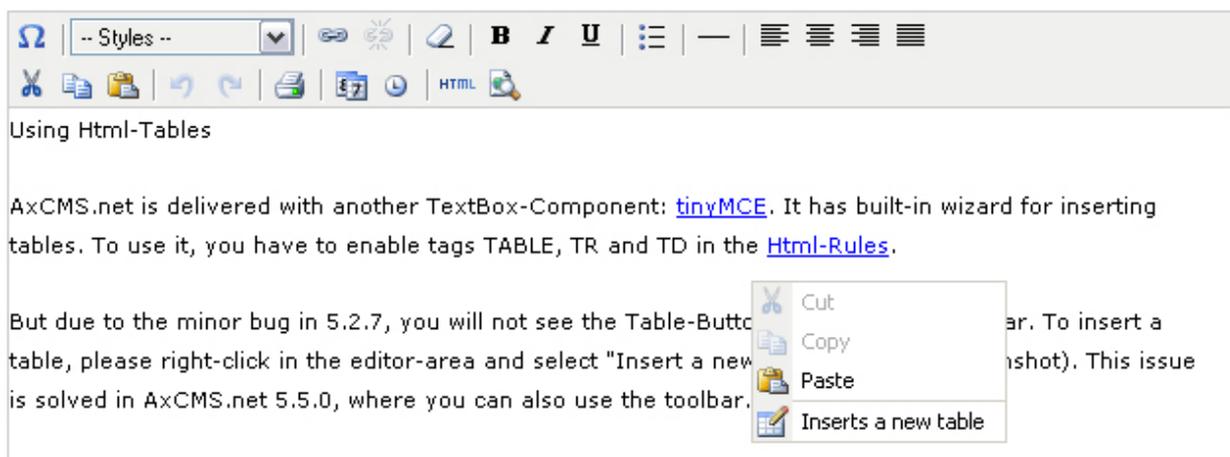
This method was unintentionally removed from 5.2.7. It will be restored back in 5.5.1. Until that you can use this workaround:

```
Classifier.ImFeelingLuckyNavigation(pageID, new int[] {(int)SubTree.Navigation})
```

Using Html-Tables

AxCMS.net is delivered with another TextBox-Component: [tinyMCE](#). It has built-in wizard for inserting tables. To use it, you have to enable tags TABLE, TR and TD in the [Html-Rules](#).

But due to the minor bug in 5.2.7, you will not see the Table-Button in the tinyMCE-toolbar. To insert a table, please right-click in the editor-area and select "Insert a new table" (see the screenshot). This issue is solved in AxCMS.net 5.5.0, where you can also use the toolbar.



Allow P-tag and CLASS-attribute for it

The new text editor uses not only SPAN-tag to apply styles, but also P-tag with a CLASS-attribute. Please allow both in the [Html-Rules](#).

Using Flexible Tables

Even if you don't allow the Editors to use Html-Tables, you can add some table-support with AxCMS.net 5.2.7 with [static custom elements](#). In the Sample-Project, delivered with AxCMS.net 5.2.7 you will find a reference implementation under CustomElements/FlexibleTables. It allows inserting tables and specifying table attributes by Editor during run-time. The elements still render themselves as Html-Tables.

AxUserDataAgent.LoadLiveUser and LoadManagementUser eliminated

Please use AxUserDataAgent.LoadUser instead. The behaviour has not changed. The field HomeSystem in the table AxUsers is ignored.

Note: If you don't have NUnit 2.1 installed and you are using Sample version 1.1x, you'll need the file nunitframework.dll for Init_Sample to work correctly. It must be located under Init_Sample folder.

You can download the file [here](#).