**Top of Form**

|  |  |  |
| --- | --- | --- |
| **[http://icelava.net/Themes/default/images/common/title.gif](http://icelava.net/)** | [icelava.net](http://icelava.net/) **why be normal?** |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Multiple app.config files for deploying to different environments **Last post 06-24-2011, 14:30 by** [**Anonymous**](http://icelava.net/members/Anonymous.aspx)**. 6 replies.**   |  |  | | --- | --- | |  | | |  | [**Next**](http://icelava.net/forums/ThreadNavigation.aspx?PostID=2920&NavType=Next) |  |  |  | | --- | --- | | **http://icelava.net/Themes/default/images/icon_post_show.gif 05-09-2008, 6:14** | [**2920**](http://icelava.net/forums/permalink/2920/2920/ShowThread.aspx#2920) |  |  |  |  | | --- | --- | --- | | * + **icelava is not online. Last active: 13 Nov 2012, 19:26**[**icelava**](http://icelava.net/members/icelava.aspx)   + **http://icelava.net/users/avatar.aspx?userid=1**   + **Top 10 Contributor**   + **Joined on 11-24-2006**   + [**Posts 1,044**](http://icelava.net/search/SearchResults.aspx?u=1&o=DateDescending) | Multiple app.config files for deploying to different environments [**Reply**](http://icelava.net/forums/AddPost.aspx?PostID=2920)[**Quote**](http://icelava.net/forums/AddPost.aspx?PostID=2920&Quote=True)   |  | | --- | | **One of the many great strides the ASP.NET team has taken to improve web developers' experience with Visual Studio came in the manifestation of the** [**Web Deployment Project**](http://msdn.microsoft.com/en-us/asp.net/aa336619.aspx)**. The most important feature, at least in my eyes, is** [**configuration section replacement**](http://weblogs.asp.net/scottgu/archive/2005/11/06/429723.aspx)**. Keeping settings meant for different environments having it all consolidated into the correct web.config just got a whole lot more efficient and less error prone. This should have been in VS .NET 2002 right at the beginning of time, but I was just overjoyed nonetheless to finally have an "official" tool for that.**  **The problem is, that only applies to web sites. We develop more than just web sites here; Windows Forms client applications, background Windows services, or just plain old command-line console programs. How are we going to continue coping with these types of projects? There are actually a variety of different (yet similar) ways development teams are practising out there. I am not too sure why the method I discovered has not been done and discussed by others before, so I thought I'd log it down first for others' benefit of reviewing and comparison with other methods.**  **Know your destination environments**  **First thing is to identify the target environments your project outputs need to be deployed at. In this scenario we are just going to take the regular Debug and Release configurations as those used by developers to build and test on their local workstations. In addition, we will just determine that there are two external environments needing deployment - SIT and UAT.**  **This calls for separate app.config files carrying appropriate settings for the respective SIT and UAT servers. We are not looking at section replacement or external source reference here. This method simply duplicates the entire app.config file and changes the settings that need to be different. As such we shall name these files SIT.config and UAT.config. Fastest way is simply to copy and paste app.config directly back into the project, and rename the files. They can then be editted for their destination environments accordingly.** [**Maintaining the uniformity of their structure should not be a problem**](http://icelava.net/forums/post/1477.aspx)**.**  **Next, the solution and output projects need to create new configuration choices with the exact same names as those given to the .config files.**  **http://farm3.static.flickr.com/2350/2478129282_4f010dd1c1.jpg?v=0**  **As with the screen shot, I have created "SIT" and "UAT" configurations for a number of Windows service projects, as well as for the solution itself. Remember that the solution configuration is independent from projects, and one can mix and match as required. That is why in the screenshot you can see a further "SIT\_deploy" configuration for the solution; the projects themselves do not have that, and continue to use their own "SIT" configuration.**  **In actuality, it is optional for the solution to have these configurations defined, but absolutely required for the projects. This will become apparent as we explore the backend mechanism that is helping us build our projects in the first place - MSBuild.**  **MSBuild**  **If you have not realised, .NET Framework 2.0 introduced MSBuild, a "brand new" build engine that can compile Visual Studio 2005 projects without Visual Studio itself. It resembles NAnt really closely, a favoured open-source build engine with the flexibility for highly customised build processes and sequences. The cool thing is 2005 project files are actually MSBuild files themselves, and learning to understand and edit their XML structure instead of relying on the visual Visual Studio properties interface will reap great power. I will not write anything further about MSBuild per se - that is too huge a topic for this post. You should refer to the MSDN site for a conceptual introduction to MSBuild and its features. What I want to achieve here is the steps to accomplish deploying the appropriate app.config file.**  **Right click on one of the output projects, and select the Unload Project option. This will bring the project "offline", and it can be right-clicked again to expose the Edit function. This allows opening the project file in XML mode.**  **[http://farm3.static.flickr.com/2019/2477317623_51818b2e53_o_d.jpg](http://www.flickr.com/photos/icelava/2477317623/)**  **Scroll all the way down. There should be two commented out MSBuild targets there. One of them is called AfterBuild. Uncomment the AfterBuild node and edit it to look like this**  **<Target Name="AfterBuild" Condition="'$(Configuration)' == 'SIT' Or '$(Configuration)' == 'UAT'"> <Delete Files="$(TargetDir)$(TargetFileName).config" /> <Copy SourceFiles="$(ProjectDir)$(Configuration).config" DestinationFiles="$(TargetDir)$(TargetFileName).config" /> </Target>**  **What this does is to executed the AfterBuild task only if the configuration is "SIT" or "UAT". What did I state previously about naming the files and configurations exactly? So after a build is complete, the regular app.config file would have been placed in the output directory, renamed as <projectname>.exe.config. We should delete that file, since it is inappropriate for the SIT or UAT server. The moniker of $(TargetDir)$(TargetFileName).config will readily locate this file in the Delete task, and remove it. After that, we have to take the appropriate .config file based on the current configuration and convert that to replace the recently-deleted $(TargetDir)$(TargetFileName).config file. That is where $(ProjectDir)$(Configuration).config comes in to locate the file sitting in our project directory, and copy it to the current output directory.**  **Once done, save and Reload the project. Attempt a build using the SIT or UAT configuration. The output should be similar to this sample pattern**  **------ Build started: Project: afterbuild, Configuration: UAT Any CPU ------ Build started 9/5/2008 18:07:53. Target CoreResGen: No resources are out of date with respect to their source files. Skipping resource generation.**  **Target \_CopyAppConfigFile: Skipping target "\_CopyAppConfigFile" because all output files are up-to-date with respect to the input files.**  **Target CopyFilesToOutputDirectory: afterbuild -> C:\Projects\experiments\afterbuild\bin\UAT\afterbuild.exe**  **Target AfterBuild: Deleting file "C:\Projects\experiments\afterbuild\bin\UAT\afterbuild.exe.config". Copying file from "C:\Projects\experiments\afterbuild\UAT.config" to "C:\Projects\experiments\afterbuild\bin\UAT\afterbuild.exe.config".**  **Inspect the new <projectname>.exe.config and see if it contains the settings defined in UAT.config or SIT.config. Of course the MSBuild target could be expanded further to carry out more elaborate activities, but the straightforward method should suffice for a great number of build needs.** | | |  |  |  | | --- | | **[View as RSS news feed in XML](http://icelava.net/forums/rss.aspx?ForumID=24&PostID=2920)** | | |  |

Aaron Seet 2003-2009

[Powered by Community Server, by Telligent Systems ](http://communityserver.org/r.ashx?1)

Bottom of Form