Resources for Visual Studio 2010 Extensibility: A reading list for Scala.NET IDE toolsmiths

© Miguel Garcia, STS, Hamburg University of Technology
http://www.sts.tu-harburg.de/people/mi.garcia

March 8, 2010

Contents

Abstract

At this point, these notes just catalog information sources that will be useful when providing IDE support for Scala on the .NET platform. This catalog facilitates identifying early on the pre-requisites for implementation work. For example, it appears that integration with the Scandcastle documentation system does not require a self-hosting compiler :-) If you notice some subtopic not being covered, please drop me a line!

1 Editor (with views)

- Visual Studio Extensibility Developer Center
  `http://msdn.microsoft.com/en-us/vsx/default.aspx`
- 2009 Ecosystem Summit Videos
  `http://msdn.microsoft.com/en-us/vsx/ee523929.aspx`
- IronPython Tools for VS should be about to be released, `http://us.pycon.org/media/2010/talkdata/PyCon2010/009/IronPython_Tools_for_Visual_Studio_Walkthrough.pdf`

2 Build system

Integration with the build facility is important as Intellisense functionality relies on the availability of assemblies.
From wikipedia `http://en.wikipedia.org/wiki/MSBuild`

> *MSBuild is a Microsoft build platform typically used in conjunction with Visual Studio. ... Since MSBuild is available as part of .NET, it is possible to build Visual Studio projects and solutions without the Visual Studio IDE installed. MSBuild is available at no cost.*
>
> *MSBuild acts on MSBuild project files which have a similar XML syntax to Apache Ant or NAnt. Even though the syntax is based upon well-defined XML schema, the fundamental structure and operation is comparable to the traditional Unix make utility: the user specifies what will be used (typically source code files) and what the result should be (typically an application), but the utility itself decides what to do and the order in which to do it.*

- The first author of "Inside the Microsoft Build Engine: Using MSBuild and Team Foundation Build" [1] has a blog[1]
- MSDN page What's new in MSBuild 4.0, `http://msdn.microsoft.com/en-us/library/ee240939(VS.100).aspx`
- MSBuild forum, `http://social.msdn.microsoft.com/forums/en-US/msbuild/threads/`
- Channel9 talks on MSBuild, `http://channel9.msdn.com/tags/MSBuild/`

2.1 Integration with Visual Studio

- MSDN page, `http://msdn.microsoft.com/en-us/library/ms171468%28VS.80%29.aspx`
- How to: Extend the Visual Studio Build Process, `http://msdn.microsoft.com/en-us/library/ms366724(VS.100).aspx`
- "Custom project systems must use IVsBuildManagerAccessor to start builds. This topic describes the reasons for this and outlines the procedure." `http://msdn.microsoft.com/en-us/library/ee891675(VS.100).aspx`

2.2 Mixed-source building

Mixed-source (say, C# and F#) in a single VS project (with or without circular dependencies) is a problem. As of 2008, not possible for the reasons discussed at `http://cs.hubfs.net/forums/thread/4848.aspx`. Workarounds (all complicating the code) mentioned. Besides "mixed-source", another term used is "interop between languages X and Y".

- An MSBuild provider for F#, `http://strangelights.com/blog/archive/2007/08/27/1593.aspx`

[1]`http://sedodream.com/default.aspx(thatbookisaboutMSBuildinVS2008however)`

2.3 Links Sink

- The MSBuild Team blog moved to become part of The Visual Studio Blog, `http://blogs.msdn.com/visualstudio/`
- Getting Started with MSBuild 4.0, `http://blogs.msdn.com/visualstudio/archive/2010/02/25/the-visual-studio-blog.aspx`
- Build Extensibility with .NET Framework 4, `http://blogs.msdn.com/visualstudio/archive/2010/02/18/build-extensibility-with-net-framework-4.aspx`
- Visual Studio 2010 Editor Samples, `http://editorsamples.codeplex.com/`
- Quan To's Visual Studio Extensibility blog, `http://blogs.msdn.com/quanto/`
- `http://channel9.msdn.com/tags/visual-studio-extensibility/`
- `http://channel9.msdn.com/tags/extension+manager/`
- Community contributed tasks, `http://msbuildtasks.tigris.org/`
- MSBuild Extension Pack, `http://msbuildextensionpack.codeplex.com/`

3 How to write a debugger

- talk VSX205: Visual Studio 2010 Debugger Extensibility `http://channel9.msdn.com/posts/VSIPMarketing/VSX205-Visual-Studio-2010-Debugger-Extensibility/`
- Visual Studio Sample Debug Engine, `http://code.msdn.microsoft.com/debugenginesample`
- "My blog (`http://blogs.msdn.com/jacdavis`) has several articles about this sample, about Visual Studio Debug Engines and about more advanced debugging topics. Other team member blogs include Gregg (`http://blogs.msdn.com/greggm`) and Jim (`http://blogs.msdn.com/jimgries`)."

Visual Studio's "Historical Debugger" (Figure 1) appears available only for the Team System version of Visual Studio 2010.

- For Java counterparts google for "Time Travel Debugging" or "Timewarp Debugging" or "Omniscient Debugging" or "Reverse Stepping"[2].

[2] `http://www.cse.buffalo.edu/jive/`

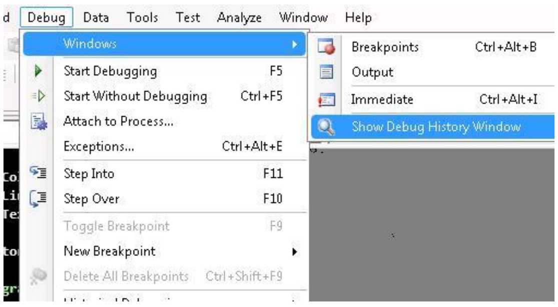

Figure 1: HistoricalDebugger

4 Playing nice with Debugging, Profiling, and Diagnostics

The talk by Mike Stall on "Debugging and the CLR", `http://compilerlab.members.winisp.net/devlab-07-debugging-handout.doc` contains useful best-practices (for an excerpt see below) as well as a collection of resources. `Reflection.Emit` has its strong points too[3]:

> *`Reflection.Emit` has full source-level debugging support and allows you associate source files with emitted code and then `Ref.Emit` will generate PDBs.*
>
> - *You can use Visual Studio's debugger, including interop-debugging support.*
> - *This works [also] for pure in-memory modules, where neither the .dll or .pdb is ever persisted to disk.*
>
> *See `http://blogs.msdn.com/jmstall/archive/2005/02/03/366429.aspx` for a detailed walkthrough of `Ref.Emit` debugging support.*

5 Playing nice with the Sandcastle documentation generator

- Blog at `http://blogs.msdn.com/sandcastle`
- Sandcastle @ Codeplex, `http://shfb.codeplex.com/`
- Sandcastle Site at MSDN,
  `http://msdn.microsoft.com/en-us/vstudio/bb608422.aspx`
- Slightly related: VSX104: Visual Studio 2010: Microsoft Help System
  `http://channel9.msdn.com/posts/VSIPMarketing/VSX104-Visual-Studio-2010-Microsoft-Help`

[3] `http://compilerlab.members.winisp.net/devlab-07-debugging-handout.doc`

6 Playing nice with Code Contracts

From `http://research.microsoft.com/en-us/projects/contracts/`:

> *Code Contracts provide a language-agnostic way to express coding assumptions in .NET programs. The contracts take the form of preconditions, postconditions, and object invariants. Contracts act as checked documentation of your external and internal APIs. The contracts are used to improve testing via runtime checking, enable static contract verification, and documentation generation.*

From `http://channel9.msdn.com/posts/Peli/Code-Contracts-in-the-IDE/`:

> *When a developer overrides a method with Contracts, the rewritter would automatically insert the pre-conditions and post-conditions in the method body. Unfortunately, the editor is not aware of that – leaving the developer confused ... Not anymore, Daryl's extension mines the Contracts on the fly and hosts them in the shiny new WPF editor of Visual Studio 2010.*

- Xml Documentation from Code Contracts for .Net, `http://channel9.msdn.com/posts/Peli/Xml-Documentation-from-Code-Contracts-for-Net/`

References

[1] Sayed Ibrahim Hashimi and William Bartholomew. *Inside the Microsoft Build Engine: Using MSBuild and Team Foundation Build.* Microsoft Press, 2009.