

CM Crossroads

the Configuration Management Community

Comprehensive CM with Razor by [Mike Gunderloy](#)

Visible Systems' Razor provides a well-integrated trio of cross-platform (Unix and Windows) configuration-management tools:

- Issues is the problem tracking system
- Versions is the version control system
- Threads is the release management system

Two things distinguish Razor from some of its competition. First, all three of these tools work closely together. It's clear that they have grown up together, rather than having been built as disparate pieces and then later lightly integrated. Second, they're all quite flexible. If you don't like the user interface in Issues, or the precise workflow within the tool suite, Razor is meant to be modified, and the manual shows you how.

Razor maintains its own databases as ASCII text files using its own custom database server, which communicates with the Razor clients (and with the Razor licensing server) via TCP/IP. The use of plain text and standard Unix utilities for management means that Razor is quite open to both customization and integration; if you need to work with issue data from an outside tool, for example, it's right there in front of you.

The starting point for most work with Razor will be the Issues application. Here's where you'll run into the customizable user interface as well. The Razor administrator can easily set up multiple issue forms, with checkboxes, freeform text-entry areas, radio buttons, date-entry fields, and so on. Razor also allows you to specify glyphs to associate with issues according to particular attributes, so that you can see patterns in a list of issues.

As you'd expect, the user can modify issues, filter and sort to find issues, and so on. There's also a powerful reporting capability here. A Reports Wizard makes it easy to build simple bar charts, HTML reports, pie charts and so on, but you can go beyond that. Not only can the administrator create custom reports but, because the reports are in fact generated by shell scripts, a "report" can do pretty much anything. For example, you could build a report to mail metrics about work done on all issues in the last week to your management team, if you were so inclined.

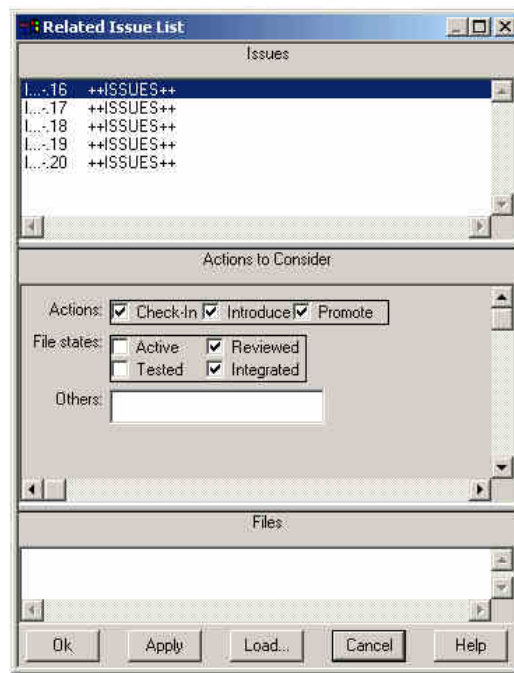
The Versions tool is layered on top of either SCCS or RCS, so if you're familiar with how those engines work you should have a good idea of what Versions can do. The major advances you'll find in Versions is that it offers a nice GUI for working with version control, and that it integrates with the rest of Razor. In particular, it's super-easy to indicate which issue a particular version control operation is meant to relate to: just drag an icon from the Issues tool and drop it in the Versions tool. Versions supports branching and merging with a reasonable though not spectacular user

interface, as well as variable promotion levels. You can also integrate your own custom scripts directly into the Versions user interface.

Threads provides a very flexible tool for release management - in fact it may provide more flexibility than many developers are used to seeing. It allows you to both define multiple releases and to capture their evolution over time, if you'd like to do that. For a simple example, say that your application consists of two files, main.c and main.h. With Threads, you could track these steps:

- On Monday, you decide version 1 of main.c and version 1 of main.h should make up version 1 of the RC thread.
- On Tuesday, you fix a bug, and version 2 of main.c and version 1 of main.h make up version 2 of the RC thread. You also define this combination as version 1 of the Release thread.
- On Wednesday, you fix another bug and version 2 of main.c and version 2 of main.h become version 2 of the Release thread.

You can define threads in many ways, from picking individual files to grabbing the most current versions of everything. You can also build a thread by grabbing all of the files that were involved in fixing a set of issues in the Issues tool. And on a larger level, you can group a thread of threads into a project. Threads and projects let you track exactly what you release at any time, and because of the close linking between the Threads, Issues, and Versions tools you get complete traceability as to what was fixed in each release and which code changes were made to fix it. If you need to, you can diff threads so you can see what's changed between different threads or different versions of a thread.



Defining a thread by choosing a list of issues and file states

Razor does not limit you to working within its native GUI clients. There's also a flexible command-line tool (named, reasonably enough, razor) for power users and scripts. A Web interface called IssueWeaver (soon to be joined by VersionWeaver) makes the Issues portion of the suite available through the browser, opening up Razor to more platforms and making it a bit more attractive to the modern GUI crowd as well. The product also supports the SCC interface for integration with tools such as Visual Studio.

As I mentioned above, Razor is cross-platform, running on Solaris, Linux, HP-UX, Irix, and AIX, and Windows, among other systems. But a word of caution is in order for those who live primarily in the Windows world: Razor shows its Unix roots quite clearly. Razor's command-line scripting relies on an installed Unix shell, and its user interface is not in line with the latest Windows styles. This shouldn't be an issue for most developers, but it's something to be aware of. On a plus side, it gives you the exact same User Interface on any platform. This means that a company will have a single learning curve for developers on multiple platforms.

There's much more to Razor than I can possibly cover in a short review. Fortunately, you can download a fully functional version from the Visible Systems Web site and try it out on your own system, either Unix or Windows. They also have extensive backup material and complete pricing information as well as the product manual available for download. You can also see an online demo or arrange for a Webinar. With Razor, you won't get the flashiest product in town, or the one with the most complex user interface or the widest variety of integrated clients. What you will get is a flexible system that makes it simple to take an integrated view of how your releases, issues, and code all relate to one another. That's more important than pretty icons in the long run.

Costs:

Razor is purchased with a floating license starting at \$815 each for 5 users excluding maintenance. IssueWeaver and VersionWeaver are sold separately. Volume discounts are available.

Company Contact:

Visible Systems
201 Spring Street
Lexington, MA 02421
Phone: +1 (781) 778-0200

sales@visible.com

www.visible.com

Mike Gunderloy, MCSE, MCSD .NET, MCDBA is an independent software consultant and author working in eastern Washington. He's the editor of ADT's Developer Central BLOG and the online Daily Grind (www.larkware.com), and the author of numerous books and articles. You can reach him at MikeG1@larkfarm.com.