



REVIEW: CONNECTION DETANGLER FOR VISUALAGE FOR SMALLTALK 4.0

Contact:
SilverMark, Incorporated
4068 Barrett Drive
Raleigh, NC 27609
<http://www.silvermark.com>
info@silvermark.com

(888)588-0668
(919) 870-7994
(919) 870-7885 (Fax)



Samuel S. Shuster is an independent consultant focusing on Smalltalk. sames@interaccess.com

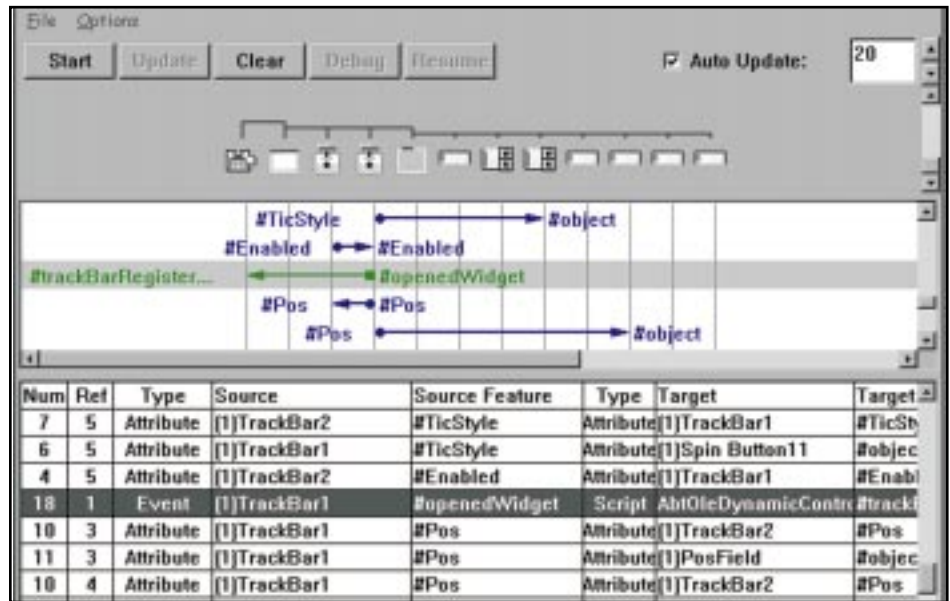


Figure 1 First and goal, inches to go.

I'M AN AVID SUPPORTER of the visual development paradigm. In other words, I'm a partysy kind of guy. That may seem like a cute simplification, but it's an important distinction. The world of high-level visual development tools is rather small. These tools are characterized by the ability to connect components (parts) that represent more than just visual elements. By contrast, trivial visual development tools do little more than paint user interfaces and allow you to attach code here and there. Nothing reasonable is available to visually represent coordinator/proposal or domain model elements. Of course, that precludes the possibility of showing how these elements interact visually.

With the introduction of PARTS, the first fully visual multipartition development tool, came the first cry of developer agony. If used recklessly, links between components in the design space start to

blur together into a tangled mess. The term *green glow* sprang up to describe this problem. Over the years, the paradigm has been discarded by the larger Smalltalk community. That is unfortunate, because when an appropriate level of discipline and rigor is used in developing in this paradigm, the problem is minimal. Unfortunately, the introduction of the VisualAge product line hasn't lessened the criticism. If anything, I await the popularization of the critical appellation *rainbow glow*.

LOOK! UP IN THE SKY!...

Into this fray comes the Connection Detangler from Silvermark. Unfortunately, the product does almost nothing for the visual "glow," regardless of how colorful it is. On hearing of this product, I hoped that somehow it would manage to untangle the connections that VisualAge poorly manages

visually. I hoped it would be a tool that would allow me to move the start and end points of the connections away from the VisualAge-imposed invisible terminator, as its cousin, PARTS, does. I hoped it would be a tool that would allow me to view selected links by function or element, as VAAssist does.

Instead, I got something I didn't expect. I got a powerful connection debugger that puts to shame all previous attempts to make a debugging tool for links and connections. I only wonder why it wasn't called Connection Debugger. Maybe that sounds too simple. Certainly the notion of a detangler peaked my interest more than a debugger might. That's unfortunate, because whatever confusion the name may cause is certainly outweighed by its usefulness and power.

Starting the Connection Detangler is straightforward. Once you have a VisualAge Composition editor open, you go to the Option menu and select Connection Detangler—it's amazing how that works! The very first time it runs, you are asked to enter registration information or confirm that you are evaluating the product. Once that's out of the way (a one-time occurrence), the Connection Detangler launches the application in the composition editor.

Sure, you say, but doesn't the Composition editor have a Connection Debugger? Yes, it does. Have you ever used it? If you're like me, you have, once, and said, "Never again." The VA Connection Debugger is a primitive, hard to decipher textual list of connections. Using the VA Connection Debugger to understand how your application works is like trying to figure out how to travel from New York to Chicago by smelling a map.

The Detangler captures all Connections, Events, Attributes, and Actions that VisualAge triggers as your application executes. The Detangler's window (see Figure 1) shows the various visual elements that are involved, a chronological trace, and an event-by-event detail table of the captured connections as they fire. As you exercise your application, the Detangler continues to capture information until you shut down the application or manually stop the Detangler. You can even choose to filter out events from selected parts or place break points on specific connections.

The information that the Detangler captures is eye-opening, if not downright disturbing. The Detangler reveals a lot of what goes on in the background at the startup of a VisualAge application. The first thing you notice is that all your attribute to attribute connections fire at least once before the application opens. That makes sense once you think it through. What isn't as easily understood is how many times

some of the connections seem to fire. I hope IBM gets a bunch of copies of this tool with an eye toward optimizing.

What can you learn from the Detangler? I'm an old hand at using PARTS, but the Detangler quickly pointed to places where my old way of thinking was wasteful. For instance, when the Detangler revealed all of those attribute-to-attribute connections firing during application start up, I realized I was duplicating effort with

some of the initialization connections I had created attached to various #opened and #aboutToOpen events.

A BIRD IN THE HAND...

Even when you don't use many connections, sometimes your application doesn't work as expected. On the day I put the Detangler through its paces, a colleague came to me with such a problem. He had a simple screen with a series of subviews. Each subview had one or two connections, and the top window had barely a handful of links. Each subform acted correctly when tested in isolation, but they didn't work right when run inside the main window. We imported his application into my image and let the Detangler loose on it. In no time, we had our answer. Simply stated, an event that occurs only once when the form was run in isolation occurs twice when run as a subview. As a result, a value was first set and then just as quickly unset. After a couple of quick changes, all was well.

Could we have figured out the problem without the Detangler? Maybe. This was one of those problems that can lead you through a dozen wild goose chases trying to trace event firings in the standard VisualAge debugger, only to end up one method call away from where you're sure the problem is, when at three in the morning in deep weariness you accidentally hit the Over instead of the Into button.

A ROSE BY ANY OTHER NAME...

Who is this tool for? Anyone who uses connections in VisualAge. Even if you're a pro and you are certain that every link you place is exactly where you want it to be, one day you'll find yourself facing behavior you didn't expect. You can either waste hours tracking it down by hand, or you can fire up your Connection Detangler and let it show you exactly where you went off course. Whether it's a set of connections that need reordering, a stray attribute being needlessly set, or a script that never seems to get called (or one that gets called too many times), you'll be hard pressed to get to the bottom of it all unless you use this tool. But then, that's what we expect from a debugger—even when they call it a Detangler. ♦

THE DETANGLER REVEALS A LOT OF WHAT GOES ON IN THE BACKGROUND AT THE STARTUP OF A VISUALAGE APPLICATION