

Revolution -- A Rapid Development Tool

David W. Brooks

Abstract

In 2005, I wrote about *Revolution* use for the back-end of a Web site and provided examples of using this tool at the core of a 24/7 practice system. The latest version of *Revolution* incorporates numerous improvements and additions. Perhaps the most powerful among these is an alpha product that will allow Web developers to plug *Revolution* files (called stacks) into Web pages and run them much like Java applets or Adobe *Flash* animations. Because I suspect that creating such *Revolution* stacks is a much easier task than creating either of the other aforementioned, chemistry educators need to know about *Revolution*. This product holds the promise of doing for us what *HyperCard* did two decades ago. We'll be able to keep our regular chemistry teaching jobs, while still having enough time to create powerful learning tools for our chemistry students.

Introduction. The last time I wrote for this Newsletter in 2005, I described **The Evolution of an Automated Practice System** that used *Revolution* as the Web back-end engine. That paper still is up and running as of 9/16/08. We teach **chemistry pedagogy courses** using this system. I also teach other courses in which I manage discussions and assignments from *Revolution*. The *Revolution* back-ends (cgi scripts) have performed extremely well for us. The only major change since my last report is that the principal server has been migrated to a high-end Macintosh server and is now physically housed in a secure bunker along with other university servers so that, when lightning strikes in Lincoln, someone other than me gets to reboot the system.

I'm writing this because of new developments that have taken place and are about to take place with *Revolution*. *Revolution* 3.0 was released in September, 2008. An enormous number of enhancements have been added since the version I wrote about in 2005. Perhaps most importantly, at the *Revolution* Conference in May, 2008, a Web browser plug-in was demonstrated that allows end-users to run *Revolution* stacks in Web pages. While not yet released as part of the *Revolution* package, that product was seeded to developers at that conference; it's much more than vaporware. Further, *Revolution's* developers are promising to add more Web back-end features. (The Mac version that I use to run my Web back-end has not been updated in years.)

A brief history. Remember *HyperCard*? Well, think of *Revolution* as *HyperCard's* grandchild. The family lines are still there, and they are strong. But, unlike Grandpa who slowly tilled the black and white electronic soil of two decades past, *Revolution* is almost as modern as you can get. It is a very powerful, almost object-oriented tool kit from which one can rapidly build complex, powerful applications -- for teaching and research. *HyperCard*, released in 1987 as part of the software package that came with every new Macintosh computer, was a sort of "software erector set." It was extensible, and you could hook up others things to it. I used it to drive videodiscs, for example. It required two screens (a bad thing), but provided extraordinary, user-history-sensitive control (a very good thing). If *HyperCard* is a "software erector set," then *Revolution* could be called *HyperCard* on steroids.

Coding. *HyperCard* was coded (programmed) in a scripting language called *HyperTalk*. *Revolution* is scripted in a language called variously *Revolution* or *TransScript*.

If you code *HyperTalk*, then you code the baby-talk version of *Revolution*. Most of your baby words are the same; you have access to a much greater adult vocabulary. In addition to more objects (such as graphics, scrollbars, and table fields), there probably are three times as many properties and commands and functions and what-have-yous in *Revolution* compared with *HyperCard*. The ability to define, set, and manipulate user-defined properties of objects is especially useful.

Perhaps what is most important is that you can get versions of *Revolution* that run on *Mac OS 9*, *Mac OS X*, several colors of *Windows*, and *Linux*. The files run cross platform. So, a file (stack) created on a *Mac* will run on a *Windows* machine. If you attend a *Revolution* conference, you'll see both platforms represented in significant numbers.

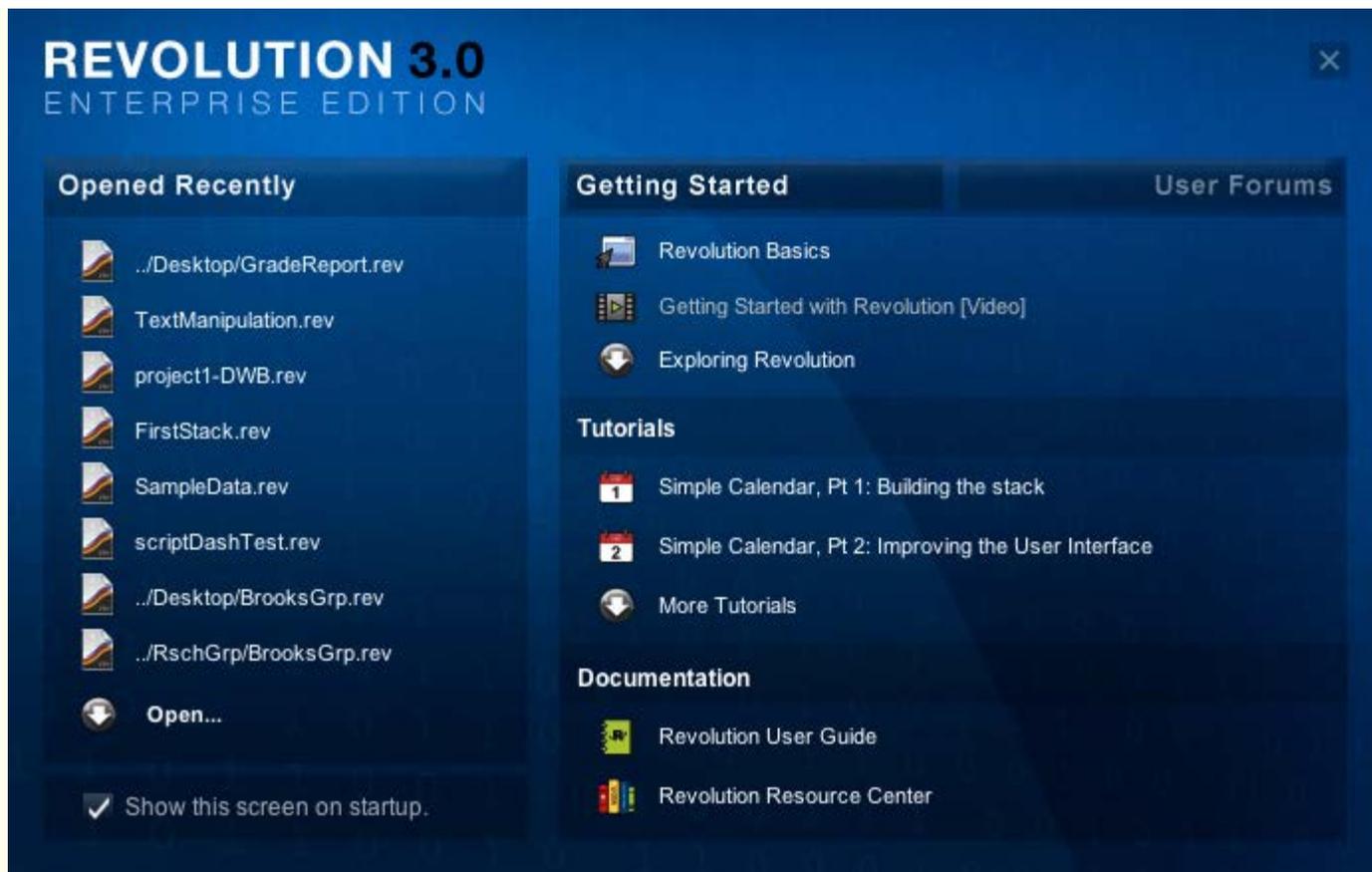
There are three levels of *Revolution*. *Media* is the bottom-end product, but does much more than *HyperCard* ever did. *Studio* is the middle version. *Enterprise* is the top version. You can make stand-alones for all platforms from any one platform with *Enterprise*, and you also have the hooks necessary to talk to *Oracle* databases. Both *Studio* and *Enterprise* are friendly to several other databases (*MySQL*, *SQLite*, *PostgreSQL*, *Valentina*).

Learning. My spouse once complained that I had bought 40 books on *HyperCard*. She was incorrect; I had bought 62. Even the 'bad' books had an example I'd not seen before. The 'good' books were filled with examples. I learned scripting by looking at and running scripts that others had written.

Those days of plentiful books are gone. There's one book on *Revolution* scripting, but it could have been written by anyone familiar with *HyperCard* and really doesn't begin to scratch the surface of those *Revolution* features unavailable in *HyperCard*. For example, one can have substacks and thereby use more than one related screen window at a time. You can have a front script that deals with events before they start up (down) the object hierarchy, and a back script that deals with them before they go to the engine.

There are numerous stacks developed in the language that can be downloaded as well as many Web-resident papers, however. When I started in 1987, there really was no WWW. Today our access to free learning materials is essentially instantaneous.

You may open *Revolution 3* with the option of this screen:



Note the listings of the Tutorials, User Guide, and Resource Center. There are many **tutorials**. Almost all come with a paired pdf document and downloadable movie. Other resources are available at the **Runtime Revolution** site.

Community. I'm quite sure I'd never have worked with *Revolution* were it not for the community of developers surrounding it. Every day my e-mail posts numerous messages from any or all of three lists. There is the 'use-revolution' list that essentially all of the major players read. They respond rapidly and courteously to even the most newbie of questions. No flaming. Then there is the 'improve-revolution' list. That's a more restricted list visited by the best among the developers -- or those willing to pay for the *Enterprise* version. That is a much more heady deal with participants reporting very nuanced bugs and asking for new features. Finally, there is a 'Revinterop' list, a really small, low-traffic group trying to establish standards for use by the entire community.

Some of the developers have created add-on products. While the latest built-in version of script editors and debuggers is really nice, there are others that are still a bit better (like **GLX2 Script Editor**). There even is a **mentoring service**. While brand new as of this writing, I suspect that the mentoring service will prove extremely useful. Some really cool games with very clever **animation** have been developed in *Revolution*. To see how relatively easy creation of a game with simple animation can be, download a trial copy of *Revolution* and build yourself a **Hangman** game.

After reading this, you might ask, "How much stock does Brooks own in Runtime Revolution (the company)?" The answer is none. *Revolution* would be replaced if I could find something better, but I haven't. I've come to realize that opensource languages like *Python* are great but need a toolkit. Universities are beginning to

place significant emphasis on opensource software and approaches, so there are subtle pushes in that direction. While opensource is conceptually sound and there are some wonderful products out there (*Moodle, Joomla, FireFox*), not all of the bases are covered. Development is a very iffy thing. While *FireFox 3* was a big improvement over *FireFox 2*, it was months before my core, everyday tools caught up.

This newsletter has been a place where those committed to utilizing computers in chemistry education can share information about developments, strategies, and products we've used successfully. *HyperCard* was a sort of miracle tool intended for a few uses but extended for many others. One of the trickiest things to learn in *HyperCard* was how to get around the 30K limit for scripts. If *HyperCard* was like a first generation tool, then *Revolution* is a third generation tool. The **University of Vienna** claims to "manage every aspect of day to day running" using *Revolution*.

I am a very satisfied user of *Revolution* and proud to consider myself a small player in its development community.