

Redmond Roundup:

Automating the Desktop

Making your job easier, one tool at a time.

November 2007 • by Peter Varhol

Your company may be growing, but the IT group isn't. There are double the number of [desktop](#) systems at your facility than there were five years ago, as well as a proliferation of notebooks, BlackBerries and other edge-computing devices. You've just added a dozen new [application](#) and file servers, and are planning on more in the next six months. Yet you have the same team you did before that growth spurt. You've cut some corners, and are using some tools for automated patch distribution and password resets, but really haven't looked into other ways of improving your processes.

It gets worse. There are more tasks to do, as well as more [systems](#) on which to do them. Security, patching and application maintenance are just a few of the tasks that require more time today. You have to make the hardware last longer, which means periodically cleaning up the systems because they became too cluttered with extraneous utilities and files. It would have saved you a great deal of effort if you could completely lock down the systems, but management insists that open access is required for business reasons.

If you're on the help desk, in desktop support, or are required to roll out new [applications](#) and patches and perform regular maintenance on applications and systems, you're looking for ways to work smarter. How can you perform the variety of everyday tasks in a way that lets you take on additional work, and additional desktops, with little or no increase in resources?

I looked at several tools that make it possible to automate processes and tasks on desktop systems. These tools are more complementary than competitive in that they perform different actions. Used separately, they can help with specific tasks that are highly time-consuming. Taken together, they can put a serious dent in your day-to-day workload.

Advanced Systems Concepts ActiveBatch

An important part of automating the desktop is the ability to execute tasks across large numbers of systems. While the scripts typically run from the server, they can perform actions on individual desktops that provide an automated way of accomplishing repetitive tasks.

ActiveBatch addresses the problem of automating those sequences of repetitive tasks across the desktops. The key to ActiveBatch is workflow. The product focuses on accomplishing sequences of processing that complete a complex task or set of tasks that can be broken down into a discrete series of steps.

I had some difficulty setting up the database necessary as a precondition to installing ActiveBatch. Rather than offering an integrated MSDE install like Desktop Authority, it required a new or existing SQL Server installation available before installing the software. It couldn't connect to my existing implementation (SQL Server 2005 Developer Edition), so I had to recreate a database connection for it to work with.

ActiveBatch lets you accomplish a variety of different things, including support for Active Directory and Windows security, the ability to perform SQL queries and search for the presence of specific files, and the ability to schedule script execution and run in a resource-constricted environment. Virtually any server or desktop task can be scripted and run automatically. The upshot is that you have to write the scripts to do the automation, but once written, they should require little or no change.

ActiveBatch	
REDMOND RATING	
Installation 20%	7.0
Features 20%	9.0
Ease of Use 20%	8.0
Administration 20%	8.0
Documentation 20%	8.0
Overall Rating:	8.0
<hr/>	
Key:	
1: Virtually inoperable or nonexistent	
5: Average, performs adequately	
10: Exceptional	

Its scripts employ variables in a hierarchical structure that allow scriptwriters to easily pass information between scripts. This means you can string scripts together to perform a series of activities while keeping them separate for flexibility or maintenance purposes.

ActiveBatch lets you compare specific changes and approvals for each revision level of a process or script to a previous or baseline change. In doing so, it identifies and documents what has changed. This provides both a change log for administrators seeking to find out what has changed and why, and an audit facility for reporting on changes for compliance purposes.

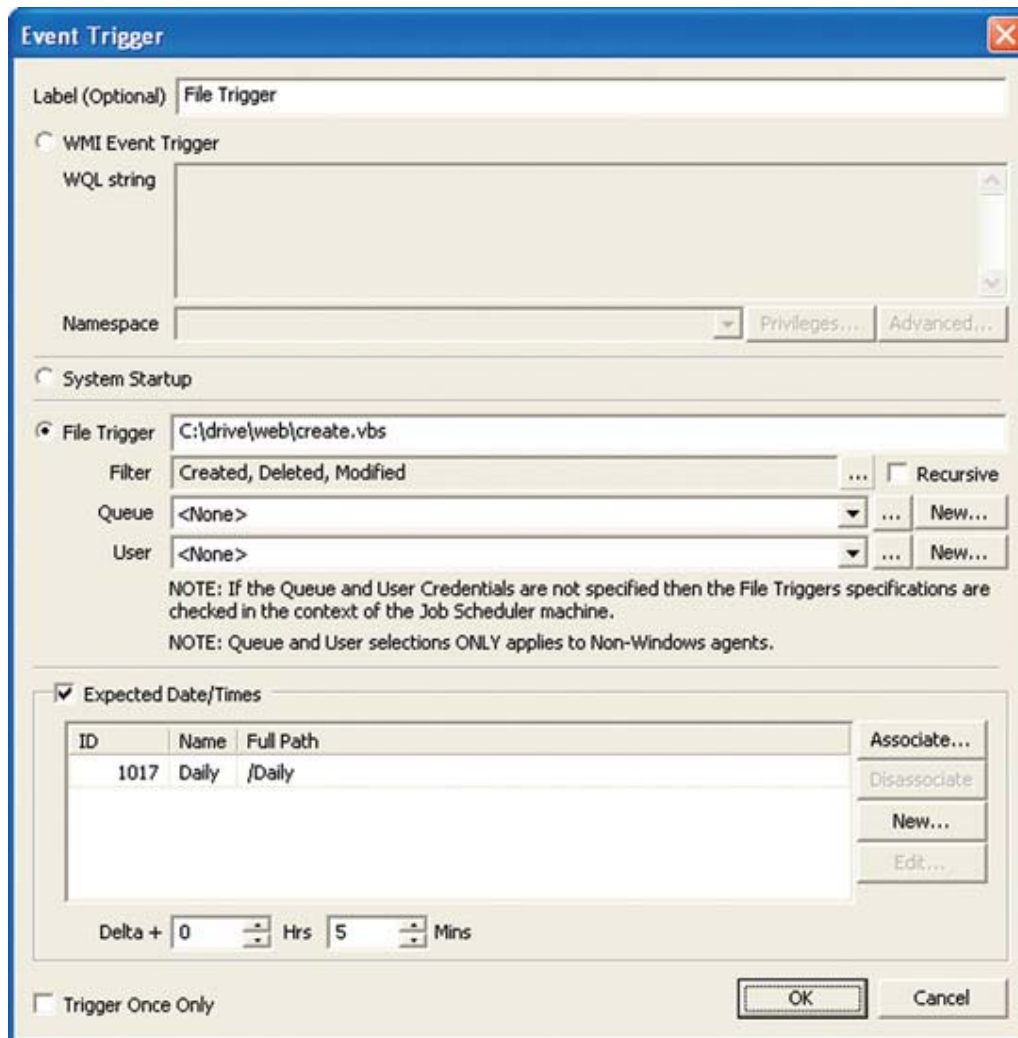


Figure 2. ActiveBatch lets you set a trigger so that if an event or sequence of events occurs, you can execute a specific response.

One of the things I really liked about ActiveBatch is the ability to design reasonably complex and complete scripts without having to actually write code. It also provides you with a set of library functions that script typical tasks, often requiring only very minor changes to work in unique environments.

Overall, if you're looking for a scripting solution that helps you design and maintain workflows, ActiveBatch will get the job done for you. That's especially true if you have a mixed server environment, as it supports Windows, Linux, AIX, Solaris, HP-UX and OpenVMS. If you're tired of maintaining dozens of Perl scripts or shell scripts to automate parts of your desktop administration, ActiveBatch can take a lot of pain out of building workflow scripts and keeping them up-to-date.

View entire article at: <http://redmondmagazine.com/features/article.asp?editorialid=2370>

Peter Varhol is the executive editor, reviews of Redmond magazine and has more than 20 years of experience as a software developer, software product manager and technology writer. He has graduate degrees in computer science and mathematics, and has taught both subjects at the university level. You can contact Peter about "Automating the Desktop" at pvarhol@redmondmag.com.