ActiveBatch
by Redwood

Expand the power of SQL server

Through intelligent IT automation

AB + Microsoft SQL Server

This paper will be a guide to reaching your fullest potential within SQL Server®

Learn how you can leverage an IT automation solution to overcome the challenges imposed by SQL Server and how you can extend the benefits of automation throughout the entire enterprise. This guide will cover:

Simplifying SQL Server Integrations

Use pre-built, proven building blocks to quickly and easily assemble workflows that incorporate and integrate SQL Server processes with other SQL Server Processes as well as non-SQL Server functions, providing end-to-end visibility and full control of your automated processes.

Cutting Batch Times and Improving Success Rates

Read how two major enterprises used intelligent automation to improve their SQL Server Scheduling, seeing results like cutting batch times by 55% and improving success rates by 216%.

Enhancing Your SQL Server Scheduling

With improved alerting & monitoring, when a problem occurs, users are not only notified of the situation, but also directed immediately to the specific workflow component in which the failure occurred. With capabilities such as References and Variables, users can save time and simplify management and maintenance throughout the entire IT environment.

How an intelligent automation solution can coordinate control, save time and resources, and extend the value of SQL Server workflows.

In enterprise IT, the sincerest form of flattery isn’t imitation;

it comes when an organization realizes that a technology is so valuable, it can no longer afford to operate in isolation.

SQL Server reached that level of importance long ago. Microsoft’s database management system is at the core of all kinds critical processes in any number of industries. And while SQL Server offers basic scheduling and workflow management tools, most IT departments will agree that those native options aren’t enough. Faster, easier, more powerful, and more innovative alternatives are needed that not only connect SQL Server to processes across their computing landscape, but also give developers considerably more control and freedom in automating complex processes.

This paper is a guide to reaching that potential for SQL Server. It begins with an outline of current SQL Server scheduling and workflow management components, then describes how, with an intelligent automation solution, new levels of productivity can be achieved for developers as well as the entire enterprise.

Native Automation in SQL Server

SQL Server users have two resident tools at their disposal to execute their database tasks. The services differ in their capabilities and purpose—but both are tried-andtrue solutions for many routine SQL Server operations. The two tools are:

SQL Server Agent

Primarily used for backups and other administrative tasks, SQL Server Agent is an automation service that initiates database management activity according to a scheduled date/time, or in response to a specific event or stored procedure.

While rudimentary in its capabilities, SQL Server Agent uses many of the same components as more sophisticated schedulers. It organizes actions into jobs that can be run on one or more local and/or remote servers. It can run more than one job on the same schedule, and multiple schedules can apply to the same job.

SQL Server Agent issues alerts when a job is completed or thresholds are met and can initiate a job in response to an alert. Alerts can be sent via email, Net Messenger, Windows Application event log, or pager; however, there is no option for popular messaging formats such as text, SNMP (Simple Network Management Protocol), or JMS (Java Message Service).

SQL Server Agent has several noteworthy limitations. If a job running under SQL Server Agent encounters a failover event, it does not resume. Jobs are logged as started, but no further completion/failure entries are sent. SQL Server Agent cannot offer failover protection via priority escalation or restarts on another machine.

Important scheduling capabilities are also lacking. There is no support for other date/ time parameters that are part of business, e.g., fiscal calendars or unique business hours. Condition-based scheduling is limited to situations such as when CPU utilization on a computer reaches an idle state. It can, however, issue calls to SQL Server Integration Services as part of a workflow.

SQL Server Integration Services (SSIS)

SQL Server Integration Services is a data migration and maintenance service that can operate both alongside and independently of SQL Server Agent. It can perform complex ETL (extract/transform/load) data services, including automated updates to multidimensional cube data.

SSIS has a rich set of built-in tasks as well as a drag-and-drop wizard for creating workflows comprised of multiple jobs and their components, commonly known as job steps. It can also store variables that are used at runtime for various purposes. Unlike SQL Server Agent, SSIS is not limited to accessing local SQL databases; it can traverse multiple endpoints on the network, performing ETL duties on flat files, relational data sources, and XML data files, among others.

SQL Server Integration Services can execute workflows across multiple SQL Server machines—a concept known as job chaining. Yet it cannot pass information from SQL Server to other servers, operating systems, or applications. Furthermore, there is no way to balance workloads across machines and systems to ensure completion.

On a larger scale, the greatest impediment to successful operations using SSIS and SQL Server Agent is the production silo in which they operate. Without automation support for both SQL Server and non-SQL Server functions, IT organizations are constrained in their ability to execute critical business tasks that are heterogeneous in nature. The restrictions of SQL Server automation tools inhibit true IT agility—and by extension, the business agility of the wider enterprise.

How Intelligent Automation Adds Value to SQL Server

Realizing the full automation potential of SQL Server jobs and workflows doesn't mean abandoning SQL Server Agent and SQL Server Integration Services. In fact, it would be counterproductive to do so. Enterprises have countless hours of effort invested in their libraries of SSIS and SQL Server Agent processes, and they need to be able to access that body of work.

To achieve a broader and more valuable vision for SQL Server automation, however, users must investigate additional technologies—specifically, intelligent IT automation. This class of software allows IT to leverage its complement of SQL Server-centric automation processes while gaining powerful new capabilities that accommodate the full technological and business landscape.

Perhaps the most valuable advantage of an intelligent automation solution is its ability to accommodate the diverse and often intricate conditions IT departments face. Event-based scheduling, for example, is far more flexible. With an automation solution, users can set parameters pertaining to business priorities, physical/virtual computing availabilities, and various software triggers. Dependencies and constraints can also be introduced to restrict or initiate execution of workflows when only specific conditions are present.

Alerting and monitoring functions are also much more versatile. When a problem occurs, users are not only notified of the situation, but also directed immediately to the specific workflow component in which the failure occurred.

As with SSIS, intelligent automation solutions support job chaining. Yet this capability is expanded significantly to include non-SQL Server, as well as SQL Server, environments. As IT organizations are increasingly required to integrate single SQL Server deployments into larger data center configurations, this capability is essential. Chaining across multiple heterogenous servers enables operators to coordinate and manage jobs without batch windows. It also allows users to incorporate non-SQL processes upstream and downstream from SQL Server processes.

From an architectural standpoint, the critical difference between SQL Server-based tools and intelligent automation is the ability to coordinate disparate technologies. Enterprises today are a mix of applications, languages, and operating systems. Within a workflow, SQL Server is often called upon to share information with a variety of other database platforms including Informatica, SAP, and/or Oracle, as well as other Microsoft applications such as Dynamics AX. The right automation solution can break down the walls surrounding SQL Server, giving organizations the agility to execute any kind of processing task, no matter how many systems and formats are involved.

Intelligent IT automation can consolidate and coordinate the full range of automation tools and tasks for maximum efficiency before, during, and after execution. Designed from the ground up for diverse environments, advanced automation solutions contain pre-tested job steps that support quick and easy design, along with easy integration to non-SQL Server platforms. They reduce the need for custom scripting, enable developers to quickly assemble end-to-end workflows, and, in real time, better manage shifting business and computing needs.

ActiveBatch
by Redwood

ActiveBatch ending silos of automation

ActiveBatch, from Advanced Systems Concepts, leads the drive for intelligent automation in SQL Server environments due to its focus on SQL Server functions and Job Steps. In doing so, it eliminates complexity and liberates developers from unnecessary and time-consuming scripting tasks.

It provides a single point of control, not only for SQL Server and other database administration jobs, but virtually any kind of enterprise computing task. Offering a layered, architectural approach, ActiveBatch is designed to simplify the changing and diverse IT landscape as well as the specific needs of SQL Server users. The platform supports SQL Server in both Windows and Linux environments, providing proven capabilities for workload automation, business processes automation, IT process automation (ITPA), managed file transfers (FTP, SFTP, etc.), big data/Hadoop automation, and much more. This unified architectural strategy is a clear improvement over the typical point-based mix of scripts and platform-specific scheduling tools that slow development, prevent high-value staff from focusing on more mission-critical activities, and prevent scalability and adaptability.

Uniquely Suited to SQL Server

ActiveBatch stretches the SQL Server automation universe through a host of production-ready, drag-and-drop integrations designed specifically for SQL Server. It enables developers to quickly assemble end-to-end workflows that address any number of SQL Server tasks.

As noted above, for any IT automation solution to truly expand the value of SQL Server, it must also respect the library of jobs and workflows already assembled using SSIS and SQL Server Agent. ActiveBatch does this, allowing SQL Server users to integrate any SSIS package or SQL Server Agent job from their existing libraries into an ActiveBatch workflow at any point—and without any additional development time. With ActiveBatch, SQL Server users can incorporate their SQL Server functions within larger workflows, pass information to SQL Server Agent jobs, and pass job parameters and data downstream from SQL Server to other jobs and systems.

ActiveBatch improves on SQL Server's automation services via a single-screen interface where SQL Server workflows can be assembled quickly and conveniently. It lets developers see and edit steps, tie SQL Server steps together, and use the results from one step in the next step. ActiveBatch provides easy access to a number of SQL Serverspecific job steps including:

- Backup (to backup a SQL Server database)
- DTS Package (executes a Data Transformation Services package for data ETL)
- SQL Server Job (executes a SQL Server job)
- T-SQLBlock (executes a TSQLBlock passing data to other applications)
- SSIS Package (executes an SSIS data migration package)

The pre-tested logic and drag-and-drop simplicity of ActiveBatch's pre-built job steps reduces custom coding dramatically. Oftentimes, developers can build and automate workflows in less than half the time it takes to write and maintain individual scripts. ActiveBatch's TSQLBlock job step is good example. TSQLBlock can be easily inserted into a workflow upstream or downstream from job steps involving other applications and processes. If a block returns a rich data set, ActiveBatch users can use that data set in a variety of ways, one example of this being a ForEachItem step. Using this step, IT has the option to pass those results downstream to a subsequent Job Step, workflow, or other process. Although this is just one example of how you can leverage this data, the ability to pass data to other applications and technologies simplifies dependency management—an otherwise time-consuming task where custom scripts would be needed. The SSIS Package job step offers another insight into ActiveBatch's SQL Server efficiencies. In addition to drawing upon all pre-existing SSIS workflows, this step lets users build in additional functionality. For example, an SSIS Package can be used to build-in and override connections to database servers in one environment (e.g., development), but sustain them when the workflow is promoted to testing, staging, or production, thereby eliminating the need to rebuild the SSIS package or manually enter a new connection each time. Developers can make use of the connection and variable overrides as well. Finally, ActiveBatch's Reference Object functionality is uniquely useful in the SQL Server environment. Reference Object, an ActiveBatch exclusive, allows hundreds or even thousands of References to mimic the same logic of an original Template job. By doing so, a single change to the Template Object will automatically be passed down to each Reference without the need for further action. Reference Objects result in faster workflow creation—but more important, they simplify maintenance of objects in the future. They make it easy to automate a maintenance process that needs to be run several times, such as cleaning up every database server in a group.

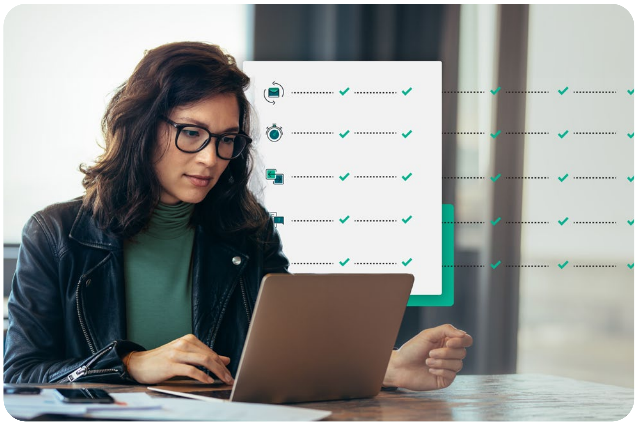
Improving SQL Server Automation with ActiveBatch®

As an intelligent workload automation solution with advantages specific to SQL Server, ActiveBatch makes it possible for developers, database administrators, IT managers, and the larger enterprise to be more efficient, doing more in less time.

Three examples illustrate how ActiveBatch speeds, simplifies, and improves SQL Server jobs:

Scheduling and Event Automation

ActiveBatch's event-driven architecture allows users to initiate or manage jobs by email, file event, message queue, database modification, tweets, and much more, as well as when specific production constraints or dependencies are met. The options, not available in SQL Server Agent, ensure that processes are triggered when specific business or IT conditions occur, thus preventing delays or process stops.



ActiveBatch scheduling capabilities can be incorporated in the execution of SQL Server jobs on the server where they are maintained for improved flexibility and performance, or on other systems running the SQL Server Client. Designers are given a choice between agent and agentless access to the SQL Server systems for invoking the SQL Server task, and incorporating these tasks into end-to-end workflows that include other systems.

ActiveBatch's authentication/user credential feature simplifies the management of jobs across multiple databases by allowing administrators to log into ActiveBatch once and execute processes across any database for which they have permissions, without the need to log-in at the server level.

Alerting

While SQL Server Agent has three alert types—SQL Server event, SQL Server performance condition, and WMI event—ActiveBatch has dozens (e.g., Job/Plan Aborted, Job/Plan Delayed in Starting, and Queue Disconnected, among others). Users can trigger jobs and workflows based on alerts prompted by unfolding conditions, without manual intervention, to remediate jobs that need assistance and proactively prevent job failures.

As a cross-platform solution, ActiveBatch unifies alerts from both SQL Server and nonSQL Server sources. This is an important benefit as it provides centralized analysis of alert conditions, offers a single point of control, optimizes workload execution, and reduces the risk of breaching SLAs.

With ActiveBatch, users have the option to receive an unlimited number of alerts via channels not available from SQL Server automation tools. The solution adds JMS, SCSM, ServiceNow, Skype and many more, ensuring that the right people receive the right alert about the right issue, at the right time, in the manner most convenient for them.

Monitoring

While Microsoft offers various monitoring tools both inside and outside of SQL Server to review and document performance, ActiveBatch offers a single source for monitoring all workflows, including those involving SQL Server. ActiveBatch monitors IT-related activity as well as workflows tied to SLAs and/or business policy. Diagnosing issues via native SQL Server monitoring tools can be a difficult task. Users are often required to drill down through as many as five different screens to understand the nature of a problem. ActiveBatch, on the other hand, can inform administrators exactly where and why a job has failed, greatly shortening the time needed to triage a problem.

ActiveBatch® Success Stories

Sub-Zero Group

Sub-Zero Group, the global manufacturer of Sub-Zero® and Wolf® brand premium appliances, uses ActiveBatch for enterprise-wide IT automation. Its uses range from giving business users self-service automation capabilities to coordinating model information and parts to vendors.

One of Sub-Zero Group's most critical ActiveBatch functions is database maintenance. In order to perform full, differential, and log backups across every SQL Server in the company, Sub-Zero's IT staff uses ActiveBatch's Reference Object capability to create three references per SQL Server. The move enables administrators to configure schedules separately, yet still do so via one job, with all reference jobs mimicking the same logic of the original template job. Any single change to the template object will be automatically passed down to each reference without the need for further action; however, reference objects can have their own triggers, constraints, alerts, and security.

> "If I want to modify that job, I just need to change it in one place instead of fifty. This not only saves us time, but it keeps maintenance consistent across our environment."
>
> Jason Van Pee Database Administrator, Sub-Sero Group

Graymont

> “The nightly failures we had to handle during our end-of-the-year budget cycle has been reduced from nearly every night to just once a month.”
>
> Paul Epp IS Manager, Greymont Services

Canadian lime and stone product producer Graymont Services was using SQL Server Agent to initiate database scripts and SQL stored procedures. It employed a collection of point scheduling tools for the end-to-end automation of ETL processes that uploaded account and financial information into a data warehouse.

Using this approach, Graymont Services had no centralized monitoring or alerting capabilities. Many times, a failed job wouldn't be discovered until the next day when a business user would complain. Furthermore, when a job failed, all other downstream jobs within a workflow would also fail. There was no capability to automatically restart a failed job; IT also lacked the ability to run jobs in parallel, and it couldn't build dependencies and constraints between jobs. Graymont Services' situation improved with the installation of ActiveBatch. Using the platform's drag-and-click interface, Paul Epp, the company's IS manager, can build workflows and manage dependencies and constraints between jobs.

He is also able to branch workflows into multiple jobs running parallel, with dependences and success/failure triggers between them, before concluding with a single “child” job. Queue Management, another ActiveBatch feature, lets Epp assign jobs across multiple servers either manually or by leveraging a generic queue that dynamically selects where the job is to be run, based on different scheduling algorithms.

Using ActiveBatch, Graymont Services is making better use of its hardware. Batch times have been cut by 55%; furthermore, due to the improved scheduling reliability of ActiveBatch and the reduced reliance on scripts, Graymont Services has seen its batch processing success rate improve from 30% to over 95%.

Microsoft SQL Server is an indispensable platform in the daily activity of a countless number of enterprises.

Its very importance is proof of the need to dependably design, build, test, deploy, and monitor database processes involving not only SQL Server itself, but also the various other platforms and technologies that make up the modern enterprise. ActiveBatch intelligent IT automation is the unified solution for this need. Its end-toend, heterogeneous compatibility, centralized control, and labor-saving design tools both enhance and simplify the use of SQL Server Agent and SQL Server Integration Services jobs and workflows. With ActiveBatch in place, SQL Server is no longer a technology silo—and IT organizations are given new agility to extend the value of their database processes to meet modern business needs.

Bring the power of ActiveBatch IT Automation to Microsoft SQL Server

activebatch.com/demo