



## Enhance Microsoft SQL Server Scheduling with ActiveBatch® Integrated Job Steps

### Benefits

- **Single Point of Scheduling** to easily integrate SQL Server jobs with other scripts and applications all managed through the ActiveBatch job scheduler.
- **Job Chaining** across multiple SQL Server systems giving operators the ability to communicate and manage multiple SQL Server jobs.
- **Pass Information** from SQL Server databases to other servers, databases, or applications.
- **Improved Reliability** through ActiveBatch non cluster failover, checkpoint restarts, and workload balancing.
- **Integrating File, Resource, and Variable Constraints** with SQL jobs, DTS and SSIS Packages, to reduce errors and ensure that data is in fact valid.
- **Customizable and Flexible Alerting** for SQL server jobs finally give organizations peace of mind knowing that their data will be up to date each and every day.

### Microsoft SQL Server: IT Boundaries Identified

With the increasing complexity of today's IT environments, businesses are looking to incorporate and integrate Microsoft SQL Server functions as a part of greater workflows that contain both SQL Server and non-SQL Server systems. Without an enterprise-wide job scheduling solution in place, system and database administrators are bounded by the limited scheduling capabilities inherent to SQL Server, for example, the ability to execute workflows across SQL Server machines (sometimes referred to as Chaining). As these jobs and dependency requirements become more complex, they also become harder to manage and the custom scripts become more time-consuming and expensive to develop and maintain. New tools are needed to easily and efficiently automate and schedule SQL Server functions as a part of greater, more complex workflows while saving time for users of SQL Server and reducing the need for custom scripting.

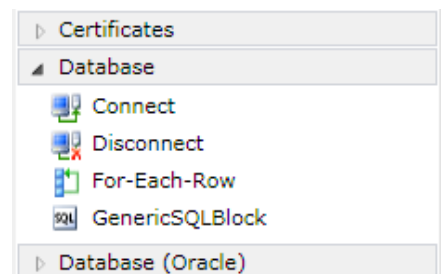
### Production-Ready Job Steps for SQL Server

ActiveBatch® Enterprise Job Scheduling and Workload Automation supports the creation of end-to-end workflows that incorporate and integrate Microsoft SQL Server functions as Job Steps within workflows. ActiveBatch provides a series of "templated", production-ready Job Steps, as an alternative to custom script creation, for a broad range of SQL Server database functions/tasks.

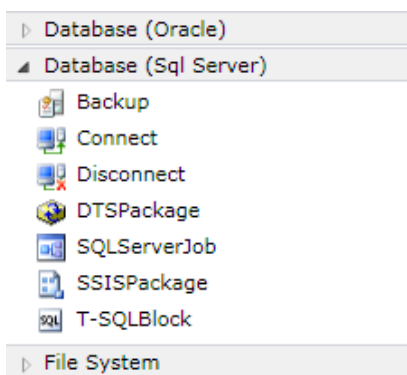
These Job Steps simplify the management of SQL Server database functions by providing auto-populating, dropdown menus and shortcuts for users to enter job properties, variables and configurations, which shortens the overall time to implementation. For example, the SSIS Job Step simplifies the management of SSIS configurations by allowing developers to use ActiveBatch Job Variables to populate SSIS configurations dynamically at runtime.

The SQL Server Job Steps are supplemented by a series of database flow control Job Steps, such as *ForEachRow*, which simplifies the implementation of workflow logic for database processes. For example, the *ForEachRow* Job Step allows a developer to designate which rows of data within the database table should be loaded or retrieved for the automation of file operations or database table cleanups. Using ActiveBatch's Integrated Jobs Library of Job Steps, IT organizations report, on average, between 50% to 80% less time building and managing SQL Server processes manually or via SQL Server Agent.

### Database Job Steps



ActiveBatch: Version 9 SP3 and above



George Bowen  
Solution Consultant  
Xcel Energy®

***"The ActiveBatch Job Step makes it much easier to manage these configurations than the built-in SSIS configuration mechanism, which is problematic to say the least. We store the configuration information in an XML file and pull that information using job variables and populate them at runtime, saving our team time and headaches."***

## Extend SQL Server Scheduling

SQL Server's scheduling functions are extended by ActiveBatch advanced scheduling capabilities that include constraints (or dependencies), monitoring, alerting, auditing and more. Job Steps within the Integrated Jobs Library allow users to incorporate operations such as FTP, SFTP, FTPS or even Web Services into workflows by coupling the business execution logic found in SQL Server with the robust framework that ActiveBatch has to offer. ActiveBatch authentication/user credentials simplifies the management of jobs across multiple databases by allowing DBAs to log into ActiveBatch once and execute processes across any database they have permissions for without the need to log-in at the server level.

ActiveBatch scheduling capabilities can be incorporated in the execution of SQL Server jobs on the server where they are maintained for the utmost flexibility and performance, or on other systems running the SQL Server Client. In addition, ActiveBatch gives the designer 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.

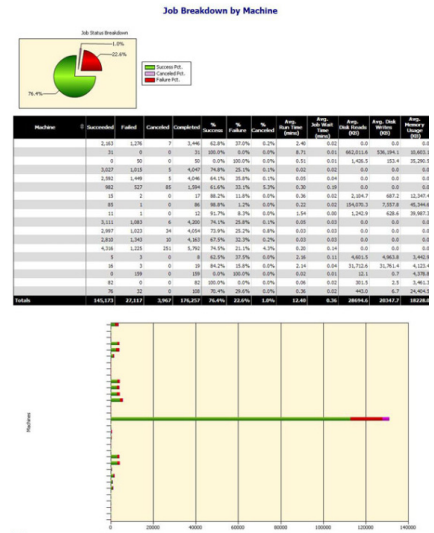
## Reporting Services

ActiveBatch supports a variety of reporting needs for today's IT organizations and users including support for packages like SQL Server Reporting, Crystal Reports and more to report on data in both list and graphical format allowing for improved analysis and better decision making

## Reporting Job Steps

- ▶ PowerShell
- ▲ Reporting
  - ▶ CrystalReport
  - ▶ CrystalReportServer
  - ▶ SqlServerReport
- ▶ System Administration

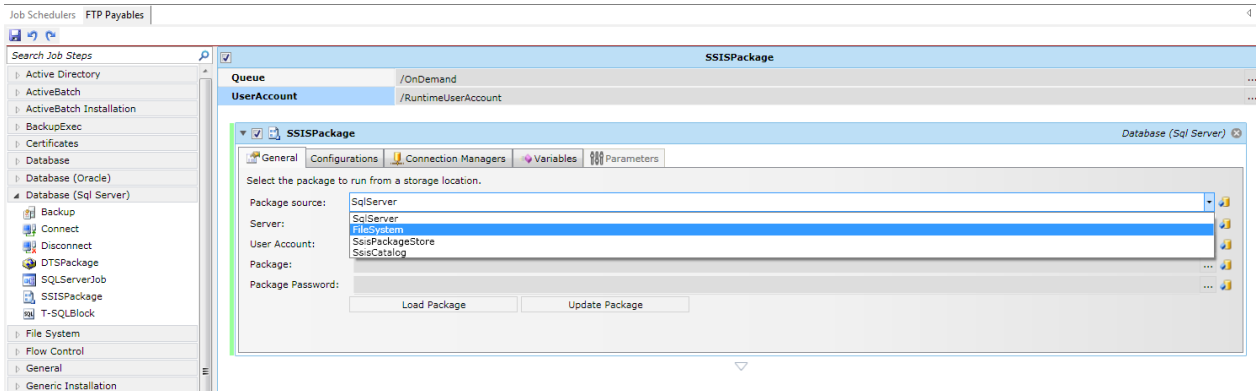
## SQL Server Reporting



## Migration Services

The ActiveBatch Migration Tool for SQL Server enables users to seamlessly convert SQL Server Agent jobs into ActiveBatch Objects without the need for manual intervention. The Migration Tool converts SQL Server jobs into T-SQL scripts, parses them and then converts them into an XML file for uploading into ActiveBatch.

## SQL Server Job Step/SSIS Package



## SQL Server Job Step/SSIS Package

