About SQL Change Automation

Overview

SQL Change Automation allows you to develop and deploy changes to a SQL Server database. It enables you to automate validation and testing, which can be performed on build and release management systems such as Azure DevOps, TeamCity and Octopus Deploy.

When developing a database, SQL Change Automation keeps track of the changes in a project folder. When it comes to release a new version, SQL Change Automation can use the project folder to produce a deployment package which can be run on the database to update it.

One way of thinking about the deployment process is 'make my database look like what's described in this SQL Change Automation project'. The state of the database is primarily represented as a set of scripts that describe how to take the database from one state to another. SQL Change Automation evaluates which scripts to run on any given database to update it. Therefore, the project folder is considered the single source of truth for how the database ought to look. This process is the same regardless of the database environment, which means deployments are repeatable and consistent.

SQL Change Automation is designed to be integrated with source control systems. A SQL Change Automation project consists of files, mainly SQL scripts, stored under a specific folder structure. Therefore, projects can be tracked with source control without any extra configuration.

SQL Change Automation is part of the SQL Toolbelt. When used alongside other SQL Toolbelt tools, such as SQL Test and SQL Data Generator, you can set up a full continuous delivery process for your database.

If you haven’t already done so, you can download the SQL Toolbelt installer.

Components

SQL Change Automation is made up of three components:

- Visual Studio Extension
  - Develop and validate database changes
- PowerShell build components
  - Builds a SQL Change Automation project to produce an artifact that can be deployed to a database
- PowerShell release components
  - Automate the deployment of the packages produced by the build components

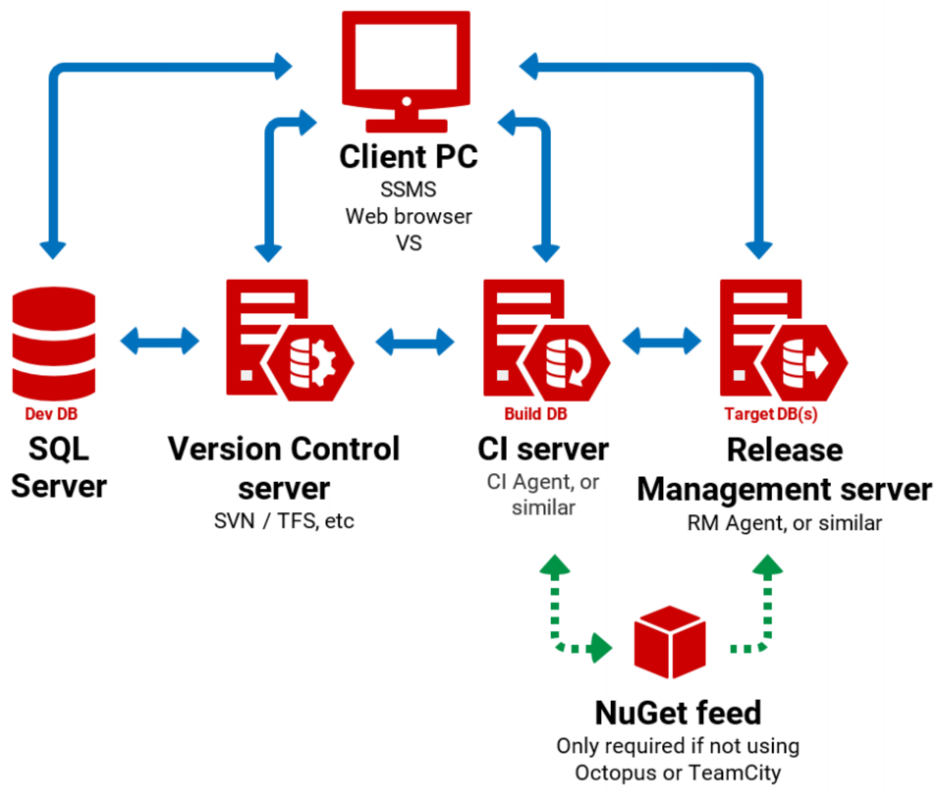
Learn more about SQL Change Automation's components.

Continuous Integration and Continuous Delivery

SQL Change Automation helps apply Continuous Integration (CI) and Continuous Delivery (CD) to database development. SQL Change Automation provides a set of PowerShell cmdlets which can automate the build and release process. There are also add-ons for some build and release systems that wrap the functionality provided by the PowerShell components.

Typical SQL Change Automation pipeline and workflow

A typical SQL Change Automation pipeline might look like this:



And a typical workflow might look like this:

- Make changes to the development database using any SQL Server management tool, for instance SQL Server Management Studio
- Capture those changes in a SQL Change Automation project using the Visual Studio extension on the Client PC
- Commit and push the SQL Change Automation project to the Version Control server using any source control client, for instance Git
- The Continuous Integration server linked to source control uses SQL Change Automation to build a deployment package
- The deployment package is transferred to the Release Management server
    - A mechanism to transfer the build artifacts from the Continuous Integration server to the Release Management server is necessary, for instance a NuGet feed
- When it's time to release, the Release Management server uses SQL Change Automation to deploy the package to a production environment

For more information see Setting Up SQL Change Automation

SQL Source Control and SQL Change Automation

It's possible to use the build and release features provided by SQL Change Automation with existing SQL Source Control projects.

For more information about how SQL Source Control relates to SQL Change Automation see SQL Change Automation projects.