

Redgate Database DevOps approaches - a comparison

| | State first approach | Migrations first approach |
|--------------|--|---|
| Summary | <p>A CREATE script representing the definition of each object in your database is stored in the version control system of your choice. Deployment scripts are automatically generated by comparing the current state of a target database with the desired database state.</p> <p>Pre- and post- deployment scripts enable flexibility for deployments where custom coding is required to effect the desired change.</p> | <p>Changes are captured in editable “migration” scripts, which are stored in the version control system of your choice. Typically, this is a series of numerically ordered ALTER scripts. In a deployment, migration scripts are applied in sequence to effect larger, iterative updates to a target database.</p> <p>State-based programmable objects and an offline schema model reduce the number of individual migration scripts needed, enable users to compare source code with databases, and support identifying merge conflicts.</p> |
| Key points | <p>Focused on defining the desired end state of your database schema, not how the transition occurs.</p> <p>A simple approach to get started with standardizing development practices, especially for teams who are experienced with SQL Compare and other state-based development tools.</p> | <p>Focused on how to get to the desired end state through precisely controlled individual migration scripts, which act as the building blocks for the eventual deployment script.</p> <p>Ideal for complex database changes and environments with high uptime requirements, as there is no ambiguity in the code which will be executed in a deployment.</p> |
| Tools needed | <p>Use the SQL Source Control extension for SQL Server Management Studio to add state to a SQL Source Control project and commit database changes to your version control system.</p> <p>Use SQL Change Automation's PowerShell cmdlets and add-ons for automation to implement Continuous Integration and automated deployments for your database changes.</p> | <p>Use the SQL Change Automation Visual Studio extension or the SQL Change Automation SQL Server Management Studio extension for database development to add changes to a SQL Change Automation project. Team members may collaborate on the same project using either or both extensions.</p> <p>Use SQL Change Automation's PowerShell cmdlets and add-ons for automation to implement Continuous Integration and automated deployments for your database changes.</p> |