Owen Priestley

Owen works as a technical writer on Redgate's SSMS plug-in SQL Source Control, producing technical copy and instructional videos. He's also lent a hand to other Redgate projects, such as DLM Dashboard, and enjoys expanding his knowledge of Redgate products whenever possible. Outside of work, Owen enjoys writing fiction, baking, playing guitar, and woodworking.

How we’ve improved migrations in SQL Source Control 5

Published 7 April 2016 10:55 am

Migration scripts, which were first introduced in SQL Source Control 3, let you assign custom SQL to specific schema changes. This custom SQL replaces the relevant part of the deployment script, which is generated by the SQL Compare engine when you perform a deployment.

With a migration script, you gain complete control over how changes are handled, which makes it possible for your teammates to get the latest changes from source control without the risk of losing data.

Why we’re taking a new approach

The first implementation of migration scripts wasn’t perfect, and a lot of our users encountered roadblocks. There was no support for Git or branching and merging, and migrations had to be linked to a separate repository location from your application code.

In 2014, we released a beta version of migrations — dubbed *Migrations V2 beta* — to address some of these problems. However, that solution was hard to use; you needed to write complex scripts, and make them compatible with any version of your database.

Feedback told us that this solution didn’t work for everyone; even the most experienced SQL developers found errors creeping into their deployments.

Instead of continuing down that path, we decided to drop the Migrations V2 beta, and improve on our original approach.

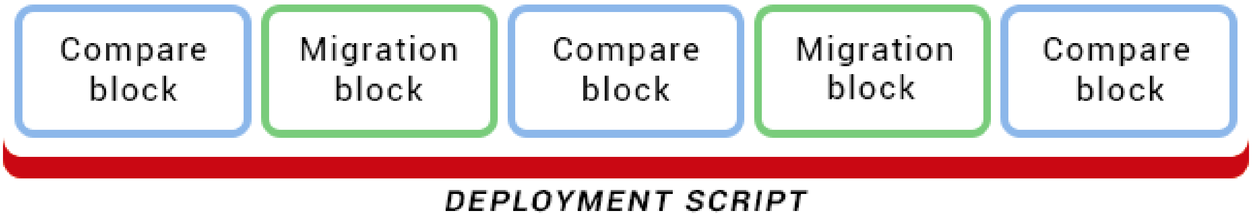
What’s actually changing?

The new migrations feature splits the deployment script into *blocks* of changes.

Regular changes, which don’t risk data loss and can be safely handled by the SQL Compare engine, are added to *compare blocks*. These blocks use the normal deployment script, which is automatically generated.

Each migration script, however, is added to its own *migrations block*. These blocks contain the SQL you or your teammates have written to handle specific changes, which could be otherwise misinterpreted by the SQL Compare engine.

Blocks are deployed in the order changes are made, so you can be sure that deployments will accurately reflect your development process.



For more information about deploying migration blocks, see how migration scripts work.

How is this an improvement?

Because changes are split into these blocks, you're given the best of both worlds: a SQL Compare deployment script by default, with the option of using migrations to manually take control of certain changes when you need to.

SQL Source Control 4 supports Git, but its current migrations solution doesn't. This was another roadblock for a lot of our users.

In SQL Source Control 5, migrations will support Git, along with branching and merging. Previous versions of migrations required the database schema to be in a specific state, which made branching and merging impossible. With the new approach, merging is fully supported, and conflict resolution is simple.

When working with states in source control, it's easy to get an overview of how the schema looks at a given time. This benefit extends to our new migrations feature — writing a migration script is easier when you know the state the schema will be in when the migration script runs.

You can also use this fine-grained control to write data-only migration scripts. This gives you the benefits of a state-based approach for all of your schema changes, while your data is handled exactly the way you want.

How this affects existing migrations users

You'll be able to automatically upgrade migration scripts created using the original migrations feature and continue to use them. However, if you're using scripts created with the Migrations V2 beta, they won't be supported after you install SQL Source Control 5. You need to deploy those scripts to all environments before you install the new beta.

Get involved

To help us build the best solution we can, we're looking for feedback from as many users as possible. We've already tested the technical solution with a private alpha, and we've recently made a public beta version available to download. That's where you come in.

Using the beta, you can create, commit and deploy migration scripts to your databases. We'll use your feedback to improve the usability of migrations and shape the future of SQL Source Control, ahead of the public release in late spring. This is your chance to really have some input on where we take these latest improvements, so don't be shy!

You can download the beta here. If you need help, or you've got any feedback for the team, please get in touch. You can also add any feature requests and suggestions to the SQL Source Control UserVoice page.

Thank this author by sharing: G+1 0

This post has been viewed 1,367 times.

Leave a Reply

You must be logged in to post a comment.