

# How to try out SQL Source Control

This page walks you through trying out SQL Source Control. You won't need to set up your own source control system, or use real databases or repositories.

You'll go through the steps of:

- adding a database to source control
- sharing the database with a (pretend) colleague
- sharing some schema changes

This walkthrough all works on your local machine, but it's pretty much the same process when you're using a remote source control repository.

## 1. Setup WidgetDev database

Instead of using your own source control repository, we'll link to an [evaluation repository](#). This is a temporary Subversion repository created on your local machine.

1. Download [this script](#). It creates a test database, WidgetDev. We'll use WidgetDev as an example database.
2. Open Management Studio and run the script.
3. In the Object Explorer, select WidgetDev.

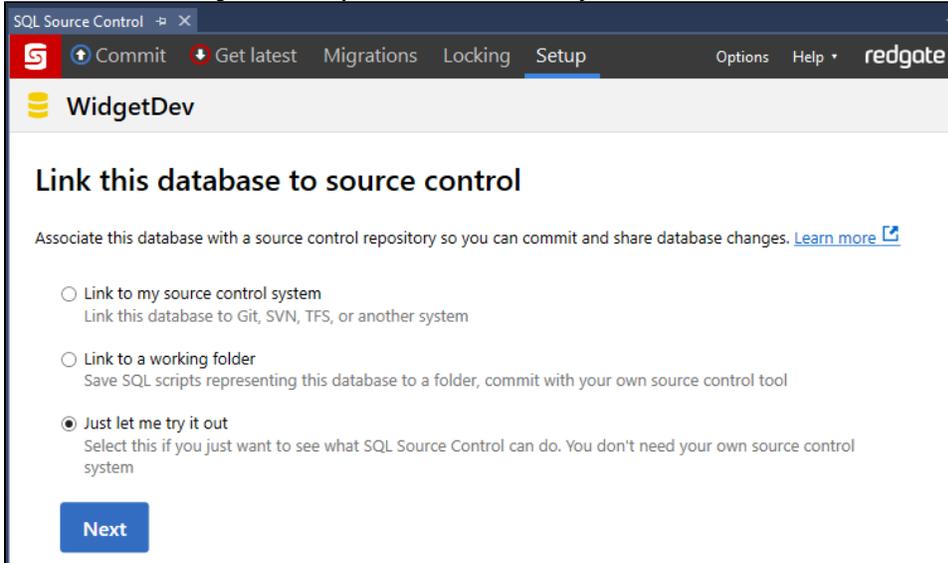
## 2. Link to a source control repository

Link the WidgetDev database to a SQL Source Control evaluation repository.

1. SQL Source Control opens as a tab when you open Management Studio. If it's not open, you can open it from the menu bar:

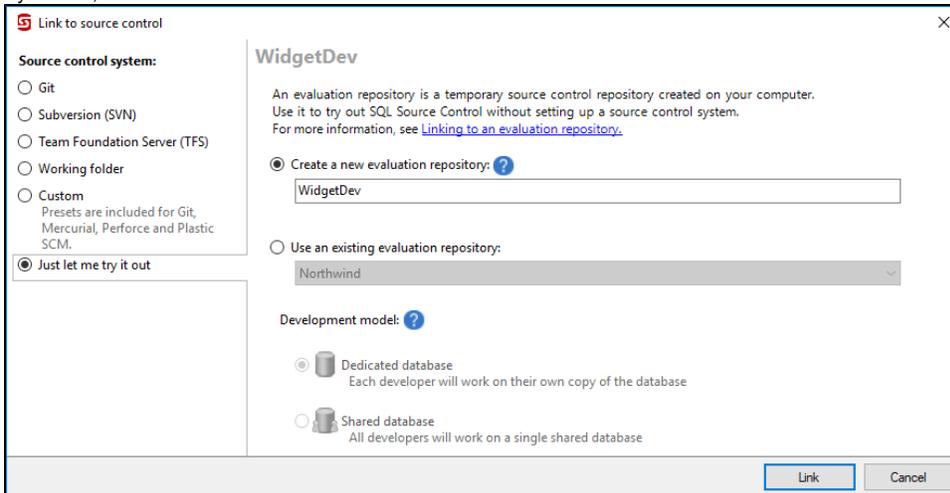


In SQL Source Control, go to the **Setup** tab, select **Just let me try it out**, and click **Next**:



## 2. Select **Create a new evaluation repository**.

By default, its name will be the name of the database:



## 3. Click **Link**.

## 3. Commit for the first time

Now you've linked WidgetDev to a source control repository, you need to **add the objects** from the database to source control by committing them. To do this:

1. In the Object Explorer, make sure WidgetDev is still selected.
2. In SQL Source Control, go to the **Commit** tab.  
This tab lists the differences between the selected database and source control. Because you haven't committed anything yet, every object in WidgetDev is listed as a new change.
3. Make sure all the changes are selected.  
When you're developing changes, you might not want to commit all your changes at the same time. For now, we want to commit everything.
4. Enter a comment - for example, "Initial commit".
5. Click **Commit**.  
SQL Source Control commits the changes to source control.

Great! WidgetDev is in source control.

If you and your team work on the same instance of a database, you can stop here. This is pretty much how you'll use SQL Source Control: just like this, everyone in the team will commit the changes they make.

However, if you use SQL Source Control to create a copy of the database for each developer, SQL Source Control becomes more powerful, as we'll see in step 3.

## 4. Create another copy of the database from source control

For this step, pretend you're another member of your database development team. We'll name him Alan.

As Alan, you'll use SQL Source Control to create your own copy of the database:

1. Create a new empty database called **WidgetDev\_Alان**.
2. In the Object Explorer, select WidgetDev\_Alان.
3. In SQL Source Control, on the **Setup** tab, select **Just let me try it out** and click **Link to source control**.
4. Click **Next**.
5. Select **Use an existing evaluation repository**.
6. From the dropdown menu, select WidgetDev and click **Link**.  
Now there are two databases linked to the same repository.
7. Go to the **Get latest** tab.  
This tab lists the differences between the selected database and the version of the database in source control. Because WidgetDev\_Alان is empty, the Get Latest tab lists every object in source control as a new change to get.
8. Click **Get latest**.  
SQL Source Control gets the changes from source control and applies them to WidgetDev\_Alان.

## 5. Make a change and share it with Alan

You and Alan have both linked copies of WidgetDev to source control. Now, you can make schema changes and share them:

1. Make a simple change to WidgetDev. For example, try adding a new table.
2. Look at the Object Explorer and you'll see a blue blob icon on WidgetDev.  
This indicates that there's an uncommitted change in the database. Expand the database to see where the change is.
3. In the Object Explorer, select WidgetDev.

4. In SQL Source Control, go to the **Commit** tab and commit the change to source control. This change is now in the repository and Alan can get it.
5. Now, pretend you're Alan again. In the Object Explorer, select WidgetDev\_Alan, and in SQL Source Control, go to the **Get latest** tab. The tab displays the change you added a moment ago.
6. Click **Get latest**. SQL Source Control gets the change from source control and applies it to WidgetDev\_Alan. Alan's database now has your changes.

You can try it the other way around too:

1. Make a simple change to WidgetDev\_Alan.
2. In SQL Source Control, go to the **Commit** tab and commit the change to source control. This change is now in the repository, and is available to your team members.
3. In the Object Explorer, select WidgetDev. Then, in SQL Source Control, go to the **Get latest** tab. The tab displays the change from Alan's database.
4. Click **Get latest**. SQL Source Control gets the change from source control and applies it to WidgetDev.

This way of working is powerful, because you and Alan can make changes independently and simultaneously. You're free to develop complex changes that might cause other parts of the database or the application to break. Once you're happy with a change, you can commit it to source control and the rest of your team can get it when they're ready.

## More things to try

There are lots more SQL Source Control features you can experiment with:

### Undo a change

If you make a change (but haven't committed it yet) and decide to undo it, you can revert the change. See [Undo a change](#) for instructions.

### See the source control history

You can see the history of all changes that have been made, along with who made them, and their comment on the change.

At the moment, this won't be very interesting because you haven't made many changes, but you can take a look anyway. See [View source control history](#).

### Get a specific version from the source control history

If you have SQL Compare, you can go back to a previous revision in source control. For example, you could go back to that initial commit you made, before your and Alan's changes.

### Filter out changes you don't want to commit

If there are objects in your database you don't want to source control - like roles - you can filter them out. See [Exclude objects using filters](#).

### Source-control static data

If you have reference data that almost never changes, you could commit this to source control. But be aware - static data can slow down SQL Source Control's performance.

## Setting up SQL Source Control with your own source control system

This example used an evaluation repository. This isn't great in the long run, because it only exists on your computer - so you can't share the repository with anyone.

You need to unlink from the evaluation repository, and [link to your own source control system](#).