

readyroll

INFOSHEET

Imperative SQL Server Database Projects
for Visual Studio 2010

info@ready-roll.com

www.ready-roll.com

[@ReadyRollSQL](https://twitter.com/ReadyRollSQL)

WHAT IS READYROLL?

ReadyRoll is a Visual Studio 2010 extension which introduces a new type of database project compatible with SQL Server 2005 (or later). ReadyRoll database projects are designed for *imperative* deployment, as opposed to the *declarative* style of Microsoft database projects. This enables fine-grain control over a database's evolution.

WHAT DOES IT DO?

ReadyRoll makes team-based development and deployment of databases simpler by providing GUI tools that integrate with the development environment. These tools allow the developer to readily switch between the logical model of a schema (in source code) and its physical model (in SQL Server).

HOW DOES IT WORK?

Visual Studio ships with a number of database project types for SQL Server built-in, from which ReadyRoll inherits its project types. A key difference being that Microsoft projects are geared primarily for an *offline* user experience: database schemas are specified as a series of SQL-based declarations and are subsequently deployed to a physical database.

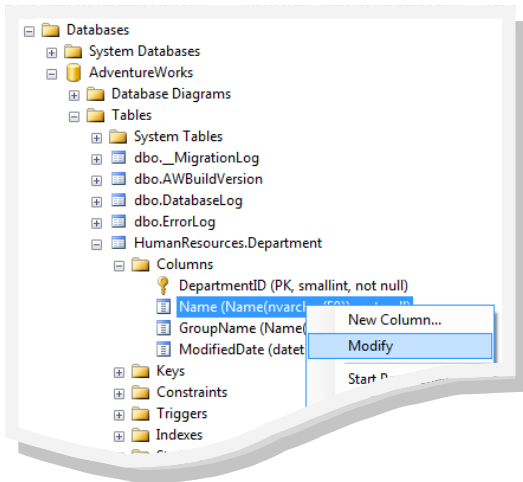
Instead, ReadyRoll favors working *online*, allowing the developer to edit their database using a tool of their choice, e.g. SQL Management Studio. ReadyRoll provides a tool-window within Visual Studio that allows the user to import database changes into their solution.

Using the ReadyRoll project *Deploy* target, those changes can then be propagated to other environments, e.g. QA or Production.



TYPICAL USE CASE

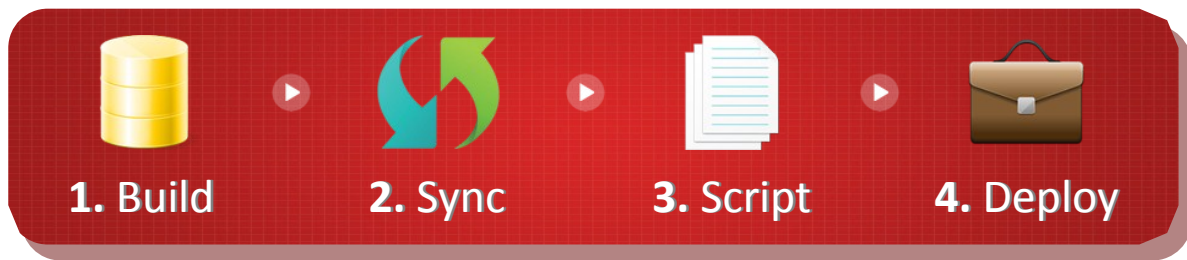
What follows is a walk-through of how a ReadyRoll project can be used to create, source control and deploy a SQL Server database.



1. BUILD

The user can start with an existing database or create a new one. In either case, the user creates a new ReadyRoll project in Visual Studio, and configures the project to point to their *Sandbox*; their local SQL Server database server (e.g. SQLEXPRESS). The first time the user performs the *Deploy* command in Visual Studio, ReadyRoll will create a versioning table in the database.

ReadyRoll uses this table to keep track of which T-SQL deployment scripts have been applied to the database.

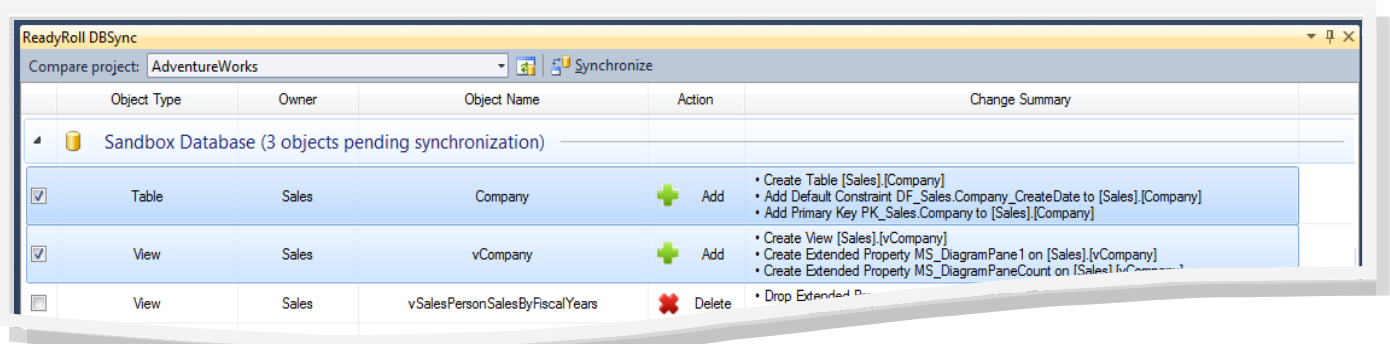


2. SYNC

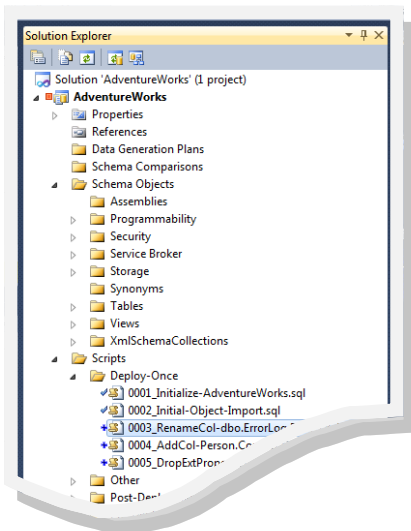
Once the user has created some objects in their database, the user switches to the ReadyRoll DBSync tool-window in Visual Studio and clicks *Start Synchronization*. This will perform a 3-way comparison between the *Sandbox* database, the Visual Studio project and a separate copy of the database called the *Shadow*.

The tool-window will display a list of pending changes, from which the user can choose to synchronize all objects or just a subset by unchecking them in the list.

During synchronization, ReadyRoll verify the referential integrity of the selected objects by executing the newly-generated script against the Shadow database.



TYPICAL USE CASE (CONTINUED)

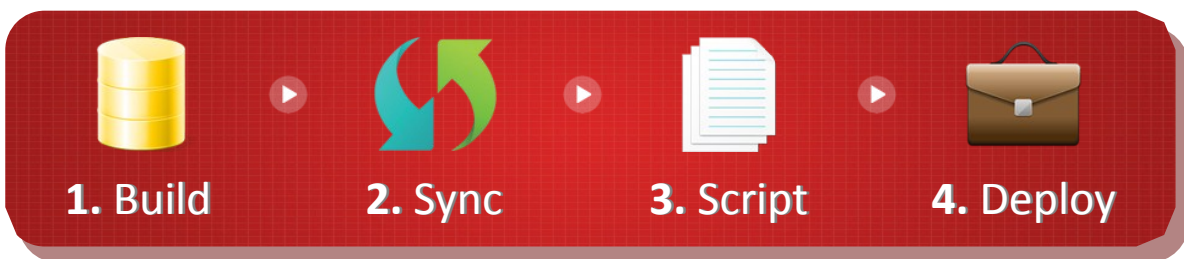


3. SCRIPT

A deployment script will be generated for the objects selected in the previous step, taking into account any interdependencies between objects.

This deployment script is added to the project under the `Scripts\Deploy-Once` sub-folder as the last item in a sequence of scripts.

The user can then edit the script to make any syntax changes, or if needed, rewrite the script to perform the change in a completely different way. For example, under certain circumstances, ReadyRoll may interpret an object rename as a DROP/CREATE operation.

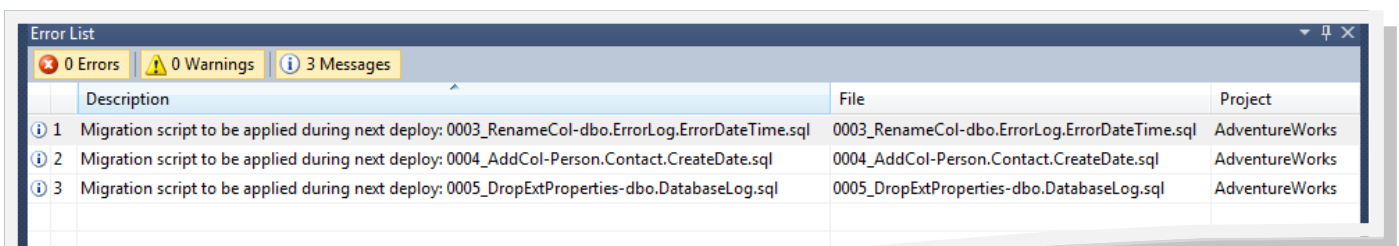


4. DEPLOY

Once the user is happy with their changes, they can elect to check their database project into source control (e.g. by using a plugin provided by the source control tool vendor).

They can then perform a project Build in Visual Studio, which will produce a T-SQL batch file containing all the deployment scripts that have been added to the project. This script ensures that each `Deploy-Once` script is executed a single time only.

This means that there is no need to generate a separate script for each environment; the same script can be used to deploy to all environments, from Development through to Production.



BENEFITS FOR YOUR TEAM

◆ **Develop with Ease**

ReadyRoll allows you to develop your database *online*: create and modify your schema directly within SQL Management Studio (or your favorite modeling tool). You can also develop your database *offline*, allowing you to search your schema and refactor objects within Visual Studio. When you're ready to synchronize your project with your database, use the ReadyRoll tool-window to preview pending changes and generate a T-SQL deployment script.

◆ **Deploy with Confidence**

Take the guesswork out of database deployments: because you define all of your deployment scripts up-front, what you build in Dev is precisely what will be deployed to Production. Just build your project each time you make a change (like you would a .NET assembly project) then use the produced SQLCMD script to deploy from Dev through to Production.

◆ **Collaborate Using Source Control**

With ReadyRoll, you can check your schema declarations and deployment scripts into source control using your current VCS provider in VS2010. This allows your team to reproduce databases at any given point of time and isolate work-in-progress in a local *Sandbox* environment. To make it easier to keep your local databases up-to-date, ReadyRoll will let you know if there are pending deployment scripts to be executed within your project.

◆ **Build for Quality**

Ensure that the integrity of your schema is maintained by utilizing the quality tools built-in to Visual Studio: write unit tests for your database, perform static code analysis, or check dependencies between schema objects. You can also add ReadyRoll projects to your MSBuild-compatible Continuous Integration server, allowing you to automate the build and deployment of your database (NOTE: VS2010 is currently required on build agents).

◆ **Roll It Your Way**

Unlock the refactoring potential of your database. Mix DDL (`CREATE`, `ALTER`, `DROP`) and DML (`SELECT`, `UPDATE`, `INSERT`) statements within your deployment scripts: add a `NOT NULL` column to a table and populate it with a value from another table, or perform a bulk data update that employs a newly-created function. ReadyRoll will keep track of each of your deployment scripts — including the order they were written — providing you with fine-grain control over how your database evolves.



PRICING

Pricing for ReadyRoll is yet to be announced, however the product will be priced competitively with other development tools.

Please contact info@ready-roll.com for more information.