SQL Packager™

Package SQL Server databases for easy installation and deployment

What is SQL Packager?

SQL Packager enables you to package the structure and contents of a Microsoft® SQL Server™ database or database upgrade into a .NET executable file or a C# project, so that installing a database is much simpler.

You can use SQL Packager to package the database structure, data, or both, for installation or deployment. SQL Packager also enables you to compress your package for reduced storage overheads, and faster deployment and distribution of databases.

It is perfect for packaging up pre-existing SQL scripts, whether generated by SQL Packager or externally. This feature makes it very simple to save out a new creation script, modify it or add comments for your client, and then simply repackage the modified SQL script.

You can use SQL Packager to package SQL Server 2008, SQL Server 2005, and SQL Server 2000 databases.

Why use SQL Packager?

When you need to deploy, install, move, or update databases on a regular basis. Packaging a database in a single executable file makes any of these tasks much simpler.

"SQL Packager is one of the most useful tools I have ever seen."

Thomas Stensitzki, iComCept GmbH

↗ SQL Packager uses a single wizard for defining the objects and data, and creating the package, so you can easily select the schema, objects, and data you want to package.

↗ You can view a plain English summary of what is being packaged.

↗ SQL Packager offers C# code templates that can be modified to suit the desired look and feel of your deployment environment.

↗ When working remotely or across multiple sites, packaging a single executable can streamline your deployment process both for whole databases and for database updates.

↗ SQL Packager has a simple, highly intuitive interface with a rich set of options for customizing your package.

How does SQL Packager work?

SQL Packager uses the powerful software engines from SQL Compare and SQL Data Compare to package your database. It is written in Visual Studio .NET and runs on the .NET framework. Databases are packaged as .NET executables that can be distributed to any machine running the .NET framework with access to the required instance of Microsoft SQL Server.

You use a simple wizard to choose schema and data, as well as to create the package. SQL Packager then packages the database by creating a .NET executable that can recreate the database, or the update of the database.

redgate®

ingeniously simple tools

SQL Packager™

When should I use SQL Packager?

Use SQL Packager when you have a database that you need to roll out, or when you need to push out changes to update databases.

Who should use SQL Packager?

SQL Packager is generally used by developers and DBAs who work with multiple SQL Server databases.

Technical specification

SQL Packager runs on a user's PC and accesses SQL Servers over the network.

Supported Environments
↗ SQL Server 2008, SQL Server 2005, and SQL Server 2000

Operating Requirements
↗ Windows 2000, Windows XP, Windows Vista or
  Windows 2003 Server
↗ SQL Server client-side tools
↗ SQL Server 2000, 2005 or 2008
↗ Microsoft .NET Framework 1.1 or 2.0
↗ MDAC 2.8+
↗ 128 MB RAM
↗ 200 MB hard disk space
↗ Runs on a user's PC (can also run on a server)

How much does SQL Packager cost?

SQL Packager costs **£195** for a single license. This provides access to all the features as well as the command line interface. SQL Packager can also be purchased in the SQL Toolbelt - a bundle of all our SQL tools representing a substantial saving over the cost of purchasing the products individually.

↗ **SQL Toolbelt™**
  SQL Compare Pro + SQL Data Compare Pro + SQL Dependency Tracker + SQL Packager + SQL Prompt Pro
  + SQL Refactor + SQL Backup Pro + SQL Data Generator + SQL Multi Script + SQL Doc Pro + SQL
  Comparison SDK = **£995**
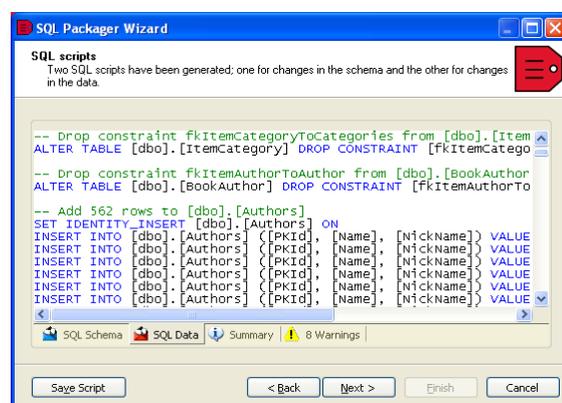
"I think it's great how it packages the changes into a single, executable file and lets you specify the target database when running that EXE. It's super-easy and should save us about 30 minutes in clean-up every time we install a new or updated database."
J. Ambrose Little, ASP.NET MVP, ASPInsider, and Web Architect, GTE Federal Credit Union

redgate®

ingeniously simple tools