The following is an excerpt from a Short Cut published by one of the Pearson Education imprints. Short Cuts are short, concise, PDF documents designed specifically for busy technical professionals like you.

We've provided this excerpt to help you review the product before you purchase. Please note, the hyperlinks contained within this excerpt have been deactivated.

**Tap into learning—NOW!**

Visit [www.informit.com/shortcuts](http://www.informit.com/shortcuts) for a complete list of Short Cuts.
RECIPE 2

Extracting Your Database in Scripts

Problem: You can’t find the right version of the database to go with your application.

You want to demonstrate your application to your customers and need the database version that goes with that application build. You have several database backups but don’t know which database version goes with which application build.

Solution

Many projects have the database structure defined in the database, which is controlled by the database administration (DBA) group. When someone wants to demonstrate the application or have another instance of the database, re-creating the database so that the application can run against it correctly is usually difficult. You want to extract the database structure into SQL scripts (text files) that you can run against an empty database so that you can have the database up and running in no time. There are several tools available for this purpose, such as Embarcadero, Toad, SQL Server Enterprise Manager, and others. If you’re starting a new project, you can begin with scripts so that you don’t have to extract them later. In addition, putting table creation scripts into one file and all other database objects into their own files makes tracking database objects easier. After you have extracted your database into scripts, make any subsequent changes to the scripts, not to the database; then you can use the scripts to re-create your database when needed.
Figure 1 shows extracting scripts from the database and checking them in to the source control repository. You have to do this task only once; as mentioned, any subsequent changes are made to the database scripts checked in to the repository, and then you check in the changes. This method gives you a better way to track changes. Later you see ways to use these scripts and how to test database creation scripts.

**FIGURE 1**
Extracting a database into scripts
**RECIPE 2**

Extracting Your Database in Scripts

**FIGURE 2**

View of the source control tree

Figure 2 shows how database creation scripts are stored in source control and how database artifacts are part of the development tree. This means you can package database artifacts with the application build. When your CI engine creates a build and tags the source control with the build number, database artifacts are also tagged with the build number. Tagging database artifacts helps you re-create the database for any build of the application.