SQLite: An SQL Database Engine In A C Library

This page was last modified on 2002/07/18 02:07:08 UTC
The latest SQLite version is **2.6.1** created on 2002/07/19 19:03:42 UTC

Introduction

SQLite is a C library that implements an embeddable SQL database engine. Programs that link with the SQLite library can have SQL database access without running a separate RDBMS process. The distribution comes with a standalone command-line access program (sqlite) that can be used to administer an SQLite database and which serves as an example of how to use the SQLite library.

SQLite is **not** a client library used to connect to a big database server. SQLite **is** the server. The SQLite library reads and writes directly to and from the database files on disk.

Features

- Implements most of SQL92.
- A complete database (with multiple tables and indices) is stored in a single disk file.
- Atomic commit and rollback protect data integrity.
- Small memory footprint: less than 20K lines of C code.
- Four times faster than PostgreSQL. Twice as fast as SQLite 1.0.
- Very simple C/C++ interface requires the use of only three functions and one opaque structure.
- TCL bindings included.
- A TCL-based test suite provides near 100% code coverage.
- Self-contained: no external dependencies.
- Built and tested under Linux and Win2K.
- Sources are uncopyrighted. Use for any purpose.

Quick Links:

- Download
- Change Log
- Report a bug

Database File Format Change - 2002 July 17

Beginning with version 2.6.0, the SQLite database file format changed in an incompatible way. If you open a database file from version 2.5.6 or earlier with version 2.6.0 or later of the library, then the file format will be converted automatically. This is an irreversible operation. Once the conversion occurs, you will no longer be able to access the database file from older versions of the library. If the database is large, the conversion might take some time. (Allow 1 to 2 seconds per megabyte of database under Linux.) If the database is read-only, the conversion cannot occur and the attempt to open the database will fail. It is suggested that you make backup copies of older database files before attempting to open them with version 2.6.0 or later of the library.

Make backups of older database files before opening them with version 2.6.0 or

later of SQLite

Current Status

A Change Summary is available on this website. You can also access a detailed change history, view open bugs, or report new bugs at the CVS server.

Complete source code and precompiled binaries for the latest release are available for download on this site. You can also obtain the latest changes by anonymous CVS access:

```
cvs -d :pserver:anonymous@cvs.hwaci.com:/home/cvs/sqlite login
cvs -d :pserver:anonymous@cvs.hwaci.com:/home/cvs/sqlite checkout sqlite
```

When prompted for a password, enter "anonymous".

Note that the CVS server is located on a cable modem with a dynamic IP address. The IP address changes every 3 or 4 months. After an IP address change occurs it usually takes a day or two for the new DNS information to propagate. So if you have trouble accessing the CVS server, it could be because the IP address has recently changed. Try again in a few days.

Whenever either of the first two digits in the version number for SQLite change, it means that the underlying file format has changed. See formatchng.html for additional information.

Documentation

The following documentation is currently available:

- Frequently Asked Questions are available online.
- Information on the sqlite command-line utility.
- The SQL Language subset understood by SQLite.
- The C/C++ Interface.
- The Tcl Binding to SQLite.
- The Architecture of the SQLite Library describes how the library is put together.
- A description of the virtual machine that SQLite uses to access the database.
- A speed comparison between SQLite, PostgreSQL, and MySQL.

The SQLite source code is 35% comment. These comments are another important source of information.

Mailing List

A mailing list has been set up on yahooGroups for discussion of SQLite design issues or for asking questions about SQLite.

Click to subscribe to sqlite

Professional Support and Custom Modifications

If you would like professional support for SQLite or if you want custom modifications to SQLite preformed by the original author, these services are available for a modest fee. For additional information contact:

D. Richard Hipp
Hwaci - Applied Software Research
704.948.4565
drh@hwaci.com

Building From Source

To build sqlite under Unix, just unwrap the tarball, create a separate build directory, run configure from the build directory and then type "make". For example:

```
$ tar xzf sqlite.tar.gz       Unpacks into directory named "sqlite"
$ mkdir bld                   Create a separate build directory
$ cd bld
$ ../sqlite/configure
$ make                        Builds "sqlite" and "libsqlite.a"
$ make test                   Optional: run regression tests
```

Related Sites

- An ODBC driver for SQLite can be found at http://www.ch-werner.de/sqliteodbc/.
- A PHP module for SQLite can be found at http://freshmeat.net/projects/sqlite-php

---

More Open Source Software from Hwaci.