

Rediscovering BerkeleyDB on the Raspberry Pi

August 29, 2014

BerkeleyDB (BDB) is an embedded database designed for in-application datastores and small devices. At its most basic, it's a simple key-value store of byte arrays.

Like [sqlite](#), the entire database is self-contained in a series of files, with no process or network port overhead.

Unlike [sqlite](#), it also has good support for managing concurrent transactions and more advanced database features such as live backups and replication.

Querying is also surprisingly advanced. Beyond hashtable style key lookups, the keys themselves are indexable and searchable, and thanks to the [secondary index feature](#), it's also possible to do complex queries within the data values themselves.

While it's possible to get a full-fledged relational dbms like [PostgreSQL running on a Raspberry Pi](#), the fact that BDB runs with minimal resources is useful in a constrained environment, especially if you have [disabled swap](#), and undertaken other [cruel and unusual measures to preserve SD card life](#) so that your Pi can run as close to 24x7x365 as possible.

Back in 2009, I was [using it extensively](#) on a messaging and workflow project, and I revisited that code when it came time to decide on a datastore for a new, long-running server application designed to run on a Pi.

I've cleaned-up that codebase and made it available as a public repo on github: <https://github.com/dpapathanasiou/pyBDB>

Archived from the original at <http://denis.papathanasiou.org/>

 Bitcoin Donate: [14TM4ADKJbaGEi8Qr8dh4KfPBQmjTshkZ2](https://blockchain.info/address/14TM4ADKJbaGEi8Qr8dh4KfPBQmjTshkZ2)