

Trac Backup

Table of Contents

- [Index](#)
- [Installation](#)
- [Upgrading](#)
- [Configuration](#)
- [Administration](#)
- [Backup](#)
- [Logging](#)
- [Permissions](#)
- [The Wiki](#)
- [Wiki Formatting](#)
- [Timeline](#)
- [Repository Browser](#)
- [Changesets](#)
- [Roadmap](#)
- [Tickets](#)
- [Ticket Queries](#)
- [Reports](#)
- [RSS Support](#)
- [Notification](#)
- [Customization](#)
- [Plugins](#)

Since Trac uses a database backend, some extra care is required to safely create a backup of a [project environment](#). Luckily, [trac-admin](#) has a command to make backups easier: `hotcopy`.

Note: Trac uses the `hotcopy` nomenclature to match that of [Subversion](#), to make it easier to remember when managing both Trac and Subversion servers.

Creating a Backup

To create a backup of a live [TracEnvironment](#), simply run:

```
$ trac-admin /path/to/projenv hotcopy /path/to/backupdir
```

[trac-admin](#) will lock the database while copying.

The resulting backup directory is safe to handle using standard file-based backup tools like `tar` or `dump/restore`.

Restoring a Backup

Backups are simply a copied snapshot of the entire [project environment](#) directory, including the SQLite database.

To restore an environment from a backup, simply stop the process running Trac (i.e. the Web server or [tracd](#)),

restore the directory structure from the backup and restart the service.

Note: Automatic backup of environments that don't use SQLite as database backend is not supported at this time. As a workaround, we recommend that you stop the server, copy the environment directory, and make a backup of the database using whatever mechanism is provided by the database system.

See also: [TracAdmin](#), [TracEnvironment](#), [TracGuide](#)