

Wikiprint Book

Title: Trac Backup

Subject: TechTIDE-Wiki - TracBackup

Version: 1

Date: 06/07/25 07:39:35

Table of Contents

Trac Backup	3
Creating a Backup	3
Restoring a Backup	3

Trac Backup

Trac backups are a copied snapshot of the entire [project environment](#) directory, including the database. Backups can be created using the `hotcopy` command of [trac-admin](#).

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

Create a backup of a live [TracEnvironment](#) by running:

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

The database will be locked while `hotcopy` is running.

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

Please note, the `hotcopy` command will not overwrite a target directory. When the target exists the operation will end with an error: `Command failed: [Errno 17] File exists:` This is discussed in [■#3198](#).

Restoring a Backup

To restore an environment from a backup, stop the process running Trac, ie the web server or [tracd](#), restore the contents of your backup to your [project environment](#) directory and restart the process.

If you are using a database other than SQLite, you'll need to restore the database from the dump file. The dump file is saved in the environment `db` directory.

To restore a PostgreSQL database backup, use the command:

```
$ psql -U <user> -d <database> -f /path/to/postgresql.dump
```

The `<database>` option is the same as the [database connection string](#) in the [\[trac\] database](#) option of `trac.ini`.

Similarly, for MySQL:

```
$ mysql -u <user> -p <database> < /path/to/mysql.dump
```

See also: [TracAdmin](#), [TracEnvironment](#), [■TracMigrate](#)