

The "Tiki Remote Instance Manager (TRIM)" will go into maintenance mode, and the code will be forked and revamped to become its replacement, now known as the "[Tiki Manager](#)".

## 1. TRIM use cases

Documentation on how to use each TRIM command resides on the [TRIM](#) page. This page is to list current and future scenarios with general steps and some tips, and example use cases.

- [1. TRIM use cases](#)
  - [1.1. Create or take over a Tiki instance](#)
  - [1.2. Backup and restore](#)
    - [1.2.1. Backups](#)
    - [1.2.2. Automated backups](#)
    - [1.2.3. Restoring](#)
  - [1.3. Automated security checks](#)
  - [1.4. Updates and upgrades](#)
    - [1.4.1. Update](#)
    - [1.4.2. Upgrade](#)
    - [1.4.3. Automated updates](#)
  - [1.5. Cloning](#)
    - [1.5.1. Manual cloning](#)
    - [1.5.2. Automated cloning](#)
  - [1.6. Test update or upgrade](#)
    - [1.6.1. Manual](#)
    - [1.6.2. Automated](#)
- [2. Future use cases](#)
  - [2.1. Re-install and apply a profile](#)
    - [2.1.1. Manual](#)
    - [2.1.2. Automated](#)
  - [2.2. Dev-testing-acceptance-prod workflow](#)
  - [2.3. Show server](#)
  - [2.4. Delayed mirroring](#)
  - [2.5. Compare Tiki instances](#)

Note: For all automated (unattended) operations, there should be a warning email upon failure (disk full or SSH failure)

### 1.1. Create or take over a Tiki instance

- `make instance`

### 1.2. Backup and restore

#### 1.2.1. Backups

- `make backup`

## 1.2.2. Automated backups

- [Set cron job](#)

## 1.2.3. Restoring

- `make restore`

## 1.3. Automated security checks

- `make watch`

## 1.4. Updates and upgrades

<b>Updates</b> from x.x to x.y	Can be automated. If you are in a branch, to the tip of that branch. If you are in trunk, you get the latest trunk.
<b>Upgrade</b> from x.x to z.z	Doesn't make sense to automate because after the 1st run, you are already at the target version.

For all updates and upgrades, a maintenance page should be shown (because weird errors can occur when part of the code has been updated)

### 1.4.1. Update

- `make backup` (to be safe)
- `make update`

### 1.4.2. Upgrade

- `make backup` (to be safe)
- `make upgrade`

### 1.4.3. Automated updates

- like `make update` but with email alert of failure

## 1.5. Cloning

### 1.5.1. Manual cloning

- `make blank` to create an instance to be cloned to
- `make clone` and select from and to

## 1.5.2. Automated cloning

Same steps as above. Then, [put on a cron job](#)

## 1.6. Test update or upgrade

### 1.6.1. Manual

- `make blank` instance to create an instance for the test upgrade
- `make clone` and select from and to
- select the version to update or upgrade to

### 1.6.2. Automated

Same steps as above. Then, [put on a cron job](#)

## 2. Future use cases

### 2.1. Re-install and apply a profile

#### 2.1.1. Manual

- `make instance` and `make profile`
- adjust your profile
- re-install Tiki and re-apply the profile
- adjust your profile until the final result is just right

#### 2.1.2. Automated

Same as above, on a daily or weekly cron job

Use cases:

- Demo sites
- Development process for complex projects with several developers
  - Thus, everyone on the team has the same data to test/develop with.
  - This is how CartoGraf was developed. Thus, the profile was applied hundreds of times before being applied in production
- Automated testing: Tests can be automated to detect regressions.

### 2.2. Dev-testing-acceptance-prod workflow

- Some features are missing in Tiki and TRIM to do this well
  - Relations between Tiki instances (This is the dev of this other one)
  - To be able to migrate data, without touching the code. Ex.: copy latest production data to the dev server (but not touching the code)

## 2.3. Show server

- <https://dev.tiki.org/show.tiki.org-Overview>

## 2.4. Delayed mirroring

Same daily operation as "Automated cloning" but with a lag (ex: one day, one week and one month). Suggested domain names include yesterday.example.org, aweekago.example.org and amonthago.example.org

Use cases:

- A user reports an issue, and indicates that current behavior is different than previous behavior. Has their been a regression in the code? A change of configuration? Is the user mistaken?
- Someone accidentally deletes a file or a tracker item, or some other operation for which their is no history (the logs indicate the action, but data is gone, so un-revertable): With aweekago.example.org, the user can find the info, and restore him/herself.

Tip: This site should be restricted (ex.: by IP or Basic authentication) because it will not have all the latest security fixes.

## 2.5. Compare Tiki instances

make compare should show a diff of all files in and outside the web directory, and a diff of database