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

Updates	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.
Upgrade	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

×

Tip

Makes sense to also set up automated backups

1.5. Cloning

Tip

The difference with a backup: A backup provides you with an archive (.tar.bz2) of your whole instance, whereas a clone is a live usable instance.

1.5.1. Manual cloning

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

×

✓ Use cases

- Debugging
- Moving Tiki to a new server
- Keep a working copy before a major upgrade. Why? After the upgrade, it may happen that 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? This site will help the process. This site should restricted (ex.: by IP or Basic authentication) because it will not have all the latest security fixes. You can give a subdomain such as 12x.example.org And when it hasn't been used in a long time, delete it.

1.5.2. Automated cloning

Same steps as above. Then, put on a cron job

×

✓ Use cases

- A fail-over instance that has data which is maximum one day old
    - If main site crashes and won't come back, then switch the DNS and the fail-over becomes the new production site. Then use other ways to restore the up-to-24 hours of lost data. Ex. from email alerts.
- A local copy in the office: Staff can continue to work if server or Internet connection is down. (keeping in mind that any changes will need to be redone on real site later
- A demo site: cloned each week from a template (reference) Tiki to show off some features.
    - If this is a site with well-known passwords, you should add extra protection to this template site (like Apache Basic authentication) so only trusted editors modify it.

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

×

✓ Use cases

- Quality Assurance (QA): Test an update or upgrade in the context of realistic data and configurations so issues can be reported early and handled before the official release of Tiki.
    - Detect regression bugs
    - Detect undesired changes
    - Detect performance issues
    - Run a manual test plan or an automated test suite
    - Test new features and thus, become familiar with them, and suggest improvements
    - Check if a bug has been fixed
        - If it's important, this indicates to the developers that a backport may be possible
        - If it can wait until the next upgrade, their is nothing more to do.

1.6.2. Automated

Same steps as above. Then, put on a cron job

×

✓ Use case

- pre-dogfood servers. Ex.: next.example.org
  - As above, but users are autonomous to check this any time and always have latest data (copied from production) and latest code.

---

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 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 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