TRIM Revamp

2018-11-25: 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".

The info below is kept for posterity.

For the foreseeable future, we'll continue working on TRIM. We did take a good look at the alternatives. Of course, they have some nice things, but they are also missing things TRIM does well. TRIM's lack of Windows support was a problem but exploring different scenarios. Ex.: TRIM on Cygwin. And Cygwin offers many other benefits as developers who are familiar with GNU/Linux can more easily work on a Windows Server.

TRIM will likely be renamed to Tiki Manager

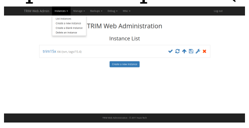
The next revamp will likely be to move from make to Symfony Console, like we already use for Tiki Console.

TRIM was started in 2008. It has a broad feature set and more are desired. It's time to think of a revamp with the following goals. See also all remaining feature requests on TRIM.

Announcements on the dev mailing list

- Tiki Remote Instance Manager (TRIM): Revamp or incremental evolution?
- Creating a ClearOS app for Tiki, TRIM Revamp, RPMs, Composer: How to manage many Tikis?

Essential

- Take a serious look at the ecosystem and evaluate if TRIM should be modernized or if we should start from and collaborate to an existing project like Rocketeer, Magallanes, Deployer, Phing, etc. ->Generic PHP app deployment tools which are written in PHP
  - We prefer a PHP-centric approach so that we can contribute
- Manage dev-staging-production workflow
  - Divergent Preferences in Staging Development Production
  - Continuous upstream
  - Configuration Management for Tiki Projects
- Be future-proof to a change in Tiki's Source Control Management. Nothing is planned but one or more of the following could happen:
  - SourceForge could shut down (Like CodePlex, Google Code, etc.)
  - SourceForge could change the URLs like they did in 2012
  - Tiki could decide to self-host with Allura, Gitlab, Phabricator, improve and use the Version Control Bridge or something else
  - Tiki could decide to move to Git, Mercurial or Fossil.
- Web GUI
  - It is understood that there will be some operations that take longer, and that some processes will need to happen or be restarted in the background. We can use some of the strategies of Tiki Scheduler
  - Proposed Bootstrap mockup GUI (by Benoit)
    - Screenshot: 

TRIM Web Administration

Instance List

trim15x tiki (svn, tags/15.4)

TRIM Web Administration - © 2017 Avan.Tech

      - *Nice! Just a little thing: I think the "List Backups" option is missing in the Backups section of the menu. Also should there be "Install/Update/Uninstall Theme" option somewhere?*

            - *I dont think TRIM can manage themes right now. As for the menu, I just listed all possible TRIM commands as of now. A lot of things will change and List Backups will probably be there eventually.*
                - Perhaps themes should be Packages? And we could enhance Profiles to fetch them? TRIM can already apply profiles.
- Windows support
    - In some corporate environments, Windows Server is the only option and it'd be better to use TRIM instead of custom scripts
    - Reference platform is Windows Server 2012, and it's OK to use something like Cygwin
        - Initial tests of TRIM on Cygwin on Windows Server 2012 ~~are promising~~ were promising but there were weird issues. We need a more stable/native solution.
    - Here are dependencies: TRIM
        - ~~SSH could be ported to PHP~~ done
- Be the Tiki app for ClearOS
    - To replace these instructions: http://wikisuite.org/How-to-install-Tiki-with-TRIM
    - So TRIM would install Tiki but also send instructions to the ClearOS API (ex.: desired PHP version, sandbox folder structure, etc.)
- Be possible in the future
    - to permit community members to run their own show server: show.tiki.org Overview
        - *+1*
    - Package in GNU/Linux distros as "Tiki Wiki CMS Groupware Manager"
        - This will bring more people to the Tiki community
        - *Not sure about the name of the package*
            - Yeah, it's a little ugly but I want people searching the repo for Wiki, CMS or Groupware to be able to find us
    - To embed in a Tiki (so a Tiki can become a management tool for other Tikis)
        - From within Tiki, TRIM could take advantage of Tiki features
            - Site list to be connected to trackers to track meta-data
                - Perhaps the show.t.o Tracker Field type can become more generic?
            - Permissions: Who is allowed to manage which Tiki
            - Billing: Credits and Payment
            - And later, to be able to evolve into http://wikisuite.org/Orchestrator **(which means orchestrating hundreds or thousands of ClearOS instances)**
                - Salt Stack is a strong candidate for this (A proof of concept was successful). So perhaps Salt Stack provisions the VM, installs and configures ClearOS and the domain name and launches TRIM to set up Tiki.
            - To be able to run automated testing on TRIM instances via Browser Automation or an external tool
    - New commands *(not sure where to put this, you can move it somewhere else)*
        - Blank instance, something like 'make blank' just to be easier/faster to make blank instances instead of 'make instance' and choosing blank instance. They are used for backup/restore/clone.
- REST API
        - It would be nice to have common Trim commands available as REST API endpoints. Let's suppose we wire Github, Sourceforge or even inotify to Trim. Whenever use change something in their source code, Trim could be informed via REST and take the proper actions, like `cd project && svn up`.
- Hooks
        - We could wire two Trim instances. Imagine we have 2 environments (Prod/Dev). In production I can configure Trim whenever a DB backup happens, tell it to other Trim in development environment, so that Dev Trim can download and load a fresh database.
- *Could it import existing Tiki installations?*

        - TRIM can "adopt" any existing Tiki which is managed by SVN. It's very important to keep this functionality. We could even go further (as a nice to have) and automate this: Convert a site installed via FTP to now use SVN

Nice to have

- FTP support: Normally, TRIM is used to managed many Tikis. You really want SSH and SVN for this. Without shell access, it's going to be a management nightmare anyway. But TRIM is useful if you have even a few Tikis. Perhaps a compromise would be for TRIM to be able to work locally on Windows. If everything is in PHP and the server has SVN for Windows, it should work OK. Could be used for backup, restore and clone. And folks can use Syncthing to get backups safely to another server.
- Better integration with Tiki's console.php (which didn't exist when TRIM started)
- Better CLI GUI
    - It's pretty cryptic now
        - *It definitely needs to be more verbose: I think the biggest need for the command line version of the server is to make it more verbose. Some commands doesn't output anything so we don't know if it was a success or not . Same thing if there are errors except when its an error generated by PHP or the OS. Commands where i wrote 'all good' means that it 'seems' to work correctly to my best of my sysadmin knowledge :O)*
    - This would be nice: http://symfony.com/doc/current/components/console/helpers/progressbar.html
        - *+1*
    - Very nice: https://github.com/php-school/cli-menu/blob/master/README.md
        - *+1*
        - *I like both of these but im wondering if its overkill to 'prettify' too much the command line version when we will have the web GUI*
- Remove some dependencies / create wrappers around command line utilities
    - Replace the usage of Make with a php console (eg. leveraging Symfony Console)
        - That also can enable the console app to show what commands are available based on the dependencies of the system
    - Replace usage of 'command line utilities' inline with the code with either PHP equivalents, or create PHP classes with a interface that allow TRIM to abstract form implementation and maybe in the future replace with a PHP based implementation, example:
        - df + awk with a directory iterator to get a directory size recursively
        - zcat with any option PHP or gzip in the command line that allow you to uncompress the file
        - rm --force with recursive unlink
        - Replace usage of ssh-copy-id with just a ssh / scp wrapper
- Pack TRIM as a Phar - that would allow to easily download trim and "start a project" using the current folder.
- Data import from existing TRIM: This should not be hard (data is in a simple sqlite db) but if it's a problem, we can live with it. We need to think of next 10 years, and as long as TRIM Revamp (Whatever the name ends up being) is able to "adopt" an existing site, we are good. It's a 1-time transition.
- TRIM was enhanced to also be able to deploy WordPress. While Tiki is the main goal, if the successor to TRIM can easily handle other PHP apps, it could be considered as the tool/base for installing PHP apps on ClearOS.

Questions

- Since https://sourceforge.net/p/tikiwiki/code/63125 , TRIM requires Composer: Are there other dependencies or checks we could manage this way?
- Keep same architecture vs use another base
    - Symfony Console (what is used in Tiki nowadays)

    - https://framework.zend.com/manual/2.4/en/modules/zend.console.introduction.html
        - *Appears it is only available for Zend Framework 2.x so far? Or just the docs?*
- *Would it support MultiTiki installs?*
    - Good question. TRIM and MultiTiki are somewhat alternatives to each other. With some extra work, we could make it work. *make backup* would be useful, but not so much *make update*. Can you elaborate on use case?

Alternatives which are specific to a web app

Major web apps typically have dedicated CLI tools

- https://github.com/hechoendrupal/drupal-console
- https://github.com/drush-ops/drush
- http://wp-cli.org/

Alternatives that are generic

Generic PHP app deployment tools which are written in PHP

Other tools

- APS installer, used by ISPconfig
- http://bldr.io
- http://robo.li
- https://laravel.com/docs/5.4/envoy
- https://taskphp.github.io
- https://github.com/ziadoz/awesome-php/blob/master/README.md#deployment

Other

- https://github.com/modess/deploying-php-applications (An e-book on the topic)
- http://www.phptherightway.com/#servers_and_deployment

Some related Tiki work

setup.sh revamp

- Move some (all?) of setup.sh to php

Relationships between various Tiki instances

In item5937 and item5930 it seems we need a new concept in TRIM to have relationships between the instances. "mirror of", "upgrade of", etc. Not sure how this should all be done. Maybe some sort of new list of 'operations' done between instance X and Y. Maybe each Tiki should "know" (ex. in db/local.php) what it is (prod / dev) and what other Tikis are related? We also want this related concept for the Federation.

System Configuration (`tiki.ini` file) is our friend to hard-code settings that make sense to the use case.
ex: dev and staging sites don't send out notification emails
So we need a standard way of setting this file where it's not overwritten in an operation.

Develop a standard methodology for staging / dev / prod for typical projects that upstream everything* except the custom theme.

- The goal is to upstream everything as per common sense:

http://community.redhat.com/blog/2015/03/upstream-first-turning-openstack-into-an-nfv-platform/
But perhaps 1 in 20 projects has an exception because of local setups / infrastructure. In these atypical setups, we usually have 1 to 5 files with some uncommitable code. We can manage manually without the overhead of a repo for this.

a) Here is a backgrounder for projects that are quite complex:
Configuration Management for Tiki Projects
When you do all this non-upstreamed code for dev/staging/prod, it can quickly become a nightmare to track it all.
b) Here are some ideas for a simple(r) process: Continuous upstream

c) Opportunity for the future (but not applicable to 12.x and older) -> Theme-related stuff (tpl, css, js, fonts, etc) can now all live in a common root directory:
https://sourceforge.net/p/tikiwiki/code/HEAD/tree/trunk/themes/fivealive/
So this could be automatically synced (rsync, Syncthing, TRIM, etc) or versioned with a different system than the Tiki (Git or Fossil). Use case: I have a production server that I can't give someone SSH access to (VPN or legal issues). But a theme integrator improves CSS on a dev server. I need to copy stuff over periodically. Syncthing could keep things in sync while keeping version history. This could be nice for tpl and css files that can both be edited via the GUI by the end users or by us as a file edit. lang/fr/custom.php is another example.

d) Recent work on Composer could also be useful. Ex.: To use Composer to deploy custom themes as Packages.