Command-line tool

~~It would be nice to have a command-line tool, similar to TRIM and Permission Check for managing all aspects of Tiki.~~ Done in Tiki11, as Tiki Console (using Symfony console)

Tasks

For now, here is the list of command lines (some are now legacy, some are still valid):

Task	current command line
Fix permissions and run Composer	sh setup.sh
Update from SVN	svn up
Update DB and clear all cache (if there are no DB changes needed, it just clears the cache)	php installer/shell.php
Install	php installer/shell.php install
Re-install and delete all data (careful!)	php installer/shell.php install
Apply a profile	php lib/profilelib/shell.php install Screencast
Update search index	php lib/search/shell.php rebuild

Missing in Tiki Console

High priority

- Backup db and files
- Restore
- Clone (backup and restore)
- Pre-dogfood server (which is in effect **clone** and **upgrade**)

Examples of commands

- Ideal: that all this can be scripted and put on cron jobs

console.php backup

- content
  - 1- db only
  - 2- db and files to a zip (like TRIM) (This is what you would put on a cron job for safe incremental backups)
    - See TRIM backups for format: TRIM
- location of files: ex.: /var/www/backups/

- name of file: enter nothing for automatically generated

Like http://www.drush.org/#archive-dump

console.php clone

- dbonly or dbandfiles (default)
- location of files: /var/www/html/11x/
- database info
    - db: Must be different than current Tiki
    - u:
    - p:
    - h: localhost is default
- upgrade?: if the clone should be upgraded, indicate version here. This is in essence a Pre-Dogfood Server like with https://sourceforge.net/p/tikiwiki/code/HEAD/tree/trunk/doc/devtools/update.dogfoodserver.sh
- redact level from Tiki DB Redactor
    - none (exact clone)
    - basic (remove stuff which are expected to be different on a another server
        - memcache_prefix
    - max if clone should be redacted, indicate here
- prefs modification. For example
    - To add a header "This is a test site, all modifications will be overriden"
    - User_watches = n (Testing changes shouldn't spam people)
    - admin password should become XYZ

Note: If files are stored to absolute paths, there will be a problem that the clone.

Related: http://www.drush.org/#core-rsync

console.php restore

This would restore a backup to the current Tiki. Note: this should perhaps only be doable on an empty database, to avoid big errors?

- location of files: /var/www/html/11x/20130707backup_tiki_abc.sql or 20130707backup_tiki_abc.zip
    - If it's a .sql, it should restore just that
    - If it's a .zip, it would contain files and db
- If files are to be restored to an absolute path, the admin should be asked which path to restore to.

Like: http://www.drush.org/#archive-restore

console.php update

Similar to the current svnup.sh

- svn upgrade

- sh setup.sh
- update db
- clean cache
- re-index

http://www.drush.org/#pm-update
http://www.drush.org/#self-update

console.php upgrade

This would do these steps:

- svn switch
- svn upgrade
- sh setup.sh
- update db
- clean cache
- re-index

console.php clone_and_update

test a minor upgrade on a copy of the data

console.php clone_and_upgrade

This is in essence a pre-dogfood server

Should we bother?

- Health check, or should this be done via external Monitoring ?
- Set admin password to XYZ (or should be interactive profile?)
- secdb/trim check

Potential Integrations

- tiki-batch_send_newsletter.php

Prerequisites

- ~~Clean up externals~~ done
- ~~Perhaps ZF2?~~ not needed

Related apps

- TRIM
- https://github.com/jlogsdon/php-cli-tools
- http://wp-cli.org/

- http://www.drush.org/ https://drupal.org/project/drush
- https://github.com/indeyets/pake
- http://framework.zend.com/manual/1.12/en/zend.tool.framework.clitool.html
- http://symfony.com/doc/2.0/components/console/introduction.html
- https://github.com/chamilo/chash

alias

- Console