Importing ticket data

Bugzilla

Ticket data can be imported from Bugzilla using the bugzilla2trac.py script, available in the contrib/ directory of the Trac distribution.

```
$ bugzilla2trac.py
bugzilla2trac - Imports a bug database from Bugzilla into Trac.

Usage: bugzilla2trac.py [options]

Available Options:
  --db <MySQL dbname>              - Bugzilla's database
  --tracenv /path/to/trac/env      - full path to Trac db environment
  -h | --host <MySQL hostname>     - Bugzilla's DNS host name
  -u | --user <MySQL username>     - effective Bugzilla's database user
  -p | --passwd <MySQL password>   - Bugzilla's user password
  -c | --clean                     - remove current Trac tickets before importing
  --help | help                    - this help info

Additional configuration options can be defined directly in the script.
```

Currently, the following data is imported from Bugzilla:

- bugs
- bug activity (field changes)
- bug attachments
- user names and passwords (put into a htpasswd file)

The script provides a number of features to ease the conversion, such as:

- PRODUCT_KEYWORDS: Trac doesn't have the concept of products, so the script provides the ability to attach a ticket keyword instead.

- IGNORE_COMMENTS: Don't import Bugzilla comments that match a certain regexp.

- STATUS_KEYWORDS: Attach ticket keywords for the Bugzilla statuses not available in Trac. By default, the 'VERIFIED' and 'RELEASED' Bugzilla statuses are translated into Trac keywords.

For more details on the available options, see the configuration section at the top of the script.

Sourceforge

Ticket data can be imported from Sourceforge using the sourceforge2trac.py script, available in the contrib/ directory of the Trac distribution.

Mantis

Mantis bugs can be imported using the attached script.

Currently, the following data is imported from Mantis:

- bugs
- bug comments
- bug activity (field changes)

Attachments are NOT imported. If you use the script, please read the NOTES section (at the top of the file) and make sure you adjust the config parameters for your environment.

mantis2trac.py has the same parameters as the bugzilla2trac.py script:

```
mantis2trac - Imports a bug database from Mantis into Trac.

Usage: mantis2trac.py [options]

Available Options:
  --db <MySQL dbname>              - Mantis database
  --tracenv /path/to/trac/env      - Full path to Trac db environment
  -h | --host <MySQL hostname>     - Mantis DNS host name
  -u | --user <MySQL username>     - Effective Mantis database user
  -p | --passwd <MySQL password>   - Mantis database user password
  -c | --clean                     - Remove current Trac tickets before importing
  --help | help                    - This help info

Additional configuration options can be defined directly in the script.
```

Other

Since trac uses a SQL database to store the data, you can import from other systems by examining the database tables. Just go into sqlite command line to look at the tables and import into them from your application.

Using a comma delimited file - CSV

See http://trac.edgewall.org/attachment/wiki/TracSynchronize/csv2trac.2.py for details. This approach is particularly useful if one needs to enter a large number of tickets by hand. (note that the ticket type type field, (task etc...) is also needed for this script to work with more recent Trac releases)