mantis2xml Documentation

Release 1.3

Richard Gomes

Sep 27, 2017

Contents

1 Quick guide for the impatient **3**
1.1 Troubleshooting 4

2 Mapping Mantis to Launchpad Bugs **5**

3 Contributing to mantis2xml **7**
3.1 Brief History 7
3.2 Design decisions 7

Code | Bugs | Forums | License | Contact

Mantis2XML is intended to ease migration from MantisBT issues to other issue management systems. It produces a XML file which can be transformed to some other format and later imported by issue managements systems.

At the moment, only migration to Launchpad Bugs is supported. In particular, a .XLST file and a shell script are provided so that the process of migrating issues to Launchpad Bugs are greatly simplified.

Note: You may also be interested on this: Launchpad Export plugin for Mantis

CHAPTER 1

Quick guide for the impatient

Create a file named `~/.mantis2xml` which contains the URL of your Mantis instance, username and password of an user with `manager` access rigths to Mantis. For example:

```
[mantis]
url: http://mantis.example.com/
user: scott
password: beam-me-up
```

Install `xsltproc` and `jing` like shown below for Debian and derivatives:

```
$ apt-get install xsltproc jing tar bzip2 -y
```

Install Bazaar and obtain mantis2xml from Launchpad:

```
$ apt-get install bzr -y
$ bzr branch lp:mantis2xml
```

Install mantis2xml, in order to make sure all dependencies are installed:

```
$ cd mantis2xml
$ python setup.py install
```

Now run the shell script which converts your Mantis instance:

```
$ ./dump_all_issues.sh
```

If you have **only one project** in your bug tracking system, the command above is possibly all you need. If you have more than one project in Mantis, please have a look at file `dump_all_issues.sh` and follow the steps explained on the top.

If everything goes well, you will see some files create in your `~/tmp/mantis/1` directory:

```
$ ls -al ~/tmp/mantis/1
issues.txt
issues.mantis.xml
issues.launchpad.xml
issues.launchpad.xml.tar.bz2
```

- `issues.txt` : brief summary of issues
- `issues.mantis.xml` : dump of all issues as they were obtained from MantisBT
- `issues.launchpad.xml` : issues transformed for being imported by Launchpad Bugs
- `issues.launchpad.xml.tar.bz2` : created after validation succeeds

Troubleshooting

If you don't see `issues.launchpad.xml.tar.bz2`, it means that the validation of `issues.launchpad.xml` failed. The errors are reported in the standard output but you can see them again entering the following command:

```
$ jing -c ./bug-export.rnc ~/tmp/mantis/1/issues.launchpad.xml
```

The most common source of troubles is possibly related to duplicated issues. Mantis supports multiple relationships between issues, including relationships of kind `duplicated of` (for duplicates) whilst Launchpad Bugs supports zero or one duplicated issue per issue. If you faced this trouble, you can either fix it by hand in the `issues.launchpad.xml` file or fix it in Mantis, making sure that you have zero or one duplicated issues per issue, and re-run `dump-all-issues.sh`.

CHAPTER 2

Mapping Mantis to Launchpad Bugs

Mantis provides fields *Steps to Reproduce* and *Additional Information* which do not have direct counterparts in Launchpad Bugs. Also, Mantis provides detailed *relationships* between issues, like:

- *duplicate of*
- *has duplicate*
- *child of*
- *parent of*
- *related to*

which are allowed to appear multiple times. Launchpad Bugs supports only *duplicate of*, which can appear zero or one times. All other relationships are not supported. In order to accomodate these differences, these fields are incorporated in field *Bug Description* in Launchpad Bugs.

Additional information:

- XSL transformation: bug-export.xslt
- XML validation: bug-export.rnc
- Launchpad Bugs Import Format

CHAPTER 3

Contributing to mantis2xml

mantis2xml was developed in a hurry, in only a couple of days, including this documentation, which consume more time to produce than one may think. There are known bugs and obviously there is certainly a lot of room for improvement.

Unfortunately, the author does not consider further developments on mantis2xml. In case you consider taking over of this project, the author will be happy to pass the token to you.

Below you see some information which may be of some relevance:

Brief History

This application is derived from sp-mantis2github, which is a Python script that talks to MantisBT via SOAP and talks to Github via API. This application does the job wonderfully well, being able to not only migrate issues to Github, but it also inserts notes in Mantis which link to the newly created issues in Github.

Design decisions

A brief proof of concept was done: I've tried to remove call to package `github` and substitute these calls by calls to package `launchpadlib`. Due to the way sp-mantis2github was written, substituting one API by another API is not a trivial task. This idea demonstrated to be "too much", due to the very little time I reserved for performing the migration I desperately need.

So, I simply removed dependencies from github and assumed that we are plenty satisfied with a XML file, which can be later transformed to Launchpad Bugs Import Format. These are the steps I've taken:

- remove dependencies from Github API. This was done either removing some code or changing function/method names so that they are never called.
- obtain the XML representation from SOAP responses which provide objects `mc_issue_getResponse`. This is what happens when command *mantis dump issue* is entered, which causes the issue to be dumped to the standard output as an XML.

- get rid of namespaces when the aforementioned XML representation is obtained. This expedient eases a lot the creation of the .XLST file which post-processes the .XML produced.
- the process of converting the XML to Launchpad Bugs Import Format is provided via a shell script, as explained above. I've actually tried to automate in Python what is currently being done in the shell script, but it crashes after some 40 issues or so are dumped. No: I'm not willing to discover why it fails. Then I simply removed the logic and kept the shell script.