

forgeTest.pl

User guide

Scott Trunkhill

scott.trunkhill@morris.com

The following explains how to use the application forgeTest.pl to test new and existing builds of forge. Templates that produce expected and reproducible output are needed. This expected output is then used to determine if future output of forge is the same as the expected output.

The first section explains where the tests are stored in the database and how each field is used. Once the final testing of this application is finished there will be a GUI to input tests but for now use TOAD to enter your test data. The second section explains how to run the tests and what happens when they are run. This includes the location of files used. The final section shows the formatting of emails that will be sent when a test fails if it was run from a cron (errors when run from the command line are sent to the screen). If you have any questions or need any help contact Scott Trunkhill. Note that this application currently does not test the command line version of forge.

Test format in database

APPTTEST schema, table TEST, currently in production database

TEST_ID - Generated from sequence. Used to identify your test. This id number will be reported in all emails notifying you of a test failure.

TEST_TYPE – Name of test type i.e. tasktracker, sports, etc. This name can be anything you want. If you want to have your own set of tests you can put your name here. There should be no spaces.

DATA – For now this is the URL call for the template and its parameters, just like you would type into a browser. When the command line test version is complete this field will contain the data to be passed on the command line if this test is a command line test. (See APP_TYPE on how to specify command line or httpd test).

EMAIL_LIST – Comma separated list of people who will receive an email when an active test is run via a cron. When a user runs the application no emails are sent, all error messages are printed to the screen.

APP_TYPE – Either “httpd” for a browser type request or “cl” for a command line test. Note “cl” is not currently supported.

STATUS – Status of your test. Note that only “ACTIVE” tests are run during the daily test run by cron. If you have a test that you are working on and want to be able to run it from the command line but not as part of the daily test set its status to “TEST”.

DATE_CREATED – Auto generated.

NAME - Name or description of test.

DATE_UPDATED – Shows the date the test was last updated.

UPDATED_BY – Used to store who the last person to update the test was.

Testing of web version of forge

The template that is created has to produce a constant output no matter when it is run. There is a property for classifieds “**DEMO**” that has a constant set of data. Sports tests could pick a date in the past and output a box score or standings page or some other set of data that will remain constant.

The application is on ej.morris.com:usr/local/bin/testing/forgeTest.pl

Usage:

```
Usage: forgeTest.pl {[-t TEST_TYPE[,TEST_TYPE, ...] | [-s STATUS] | [-n  
NAME OR ID]} [-cd]
```

```
    -t Type of tests to run. Can be combined with -s status.  
    -s Status of tests to run. Can be combined with -t type.  
    -n ID of test to run.  
    -c Run as cron. Emails scott.trunkhill@morris.com and does not  
print results to screen.  
    -d debug  
    -h usage help.
```

You need to make sure the user running it is in eng group on ej.morris.com

The first time a test is run a baseline file is created. This is simply the output of the template. The file is stored in `/var/local/testing/forge/baseline/` with the `TEST_ID` as the file name. Then each additional time the test is run the resulting output is compared to this baseline file. If the two match the test passes and nothing is output.

If the baseline and the current output do not match the test fails and either an error message is printed to the screen (if run via command line) or an email is sent (if run from my cron).

The current output is then saved to `/var/local/testing/forge/current/YYYYMonthDD-username/` where `YYYY` is the current year, `Month` is the current month and `DD` is the current day and `username` is the username of the person running the application. In the case of crons the directory created would be `2002December17-trunk` if it was run on Dec 17, 2002. All crons will be run as user trunk. Today’s output from the test will be stored in this directory with the `TEST_ID` as the file name. You can run `diff` on the baseline file and the current file to see what has changed.

If it is later determined that the current baseline file is incorrect it can be deleted and a new file will be created the next time that test is run.

Sample Email

Here is a sample of the email sent and its format:

```
File sizes do not match -- test number 83  
URL tested http://classifieds.morris.com/classifieds-bin/classifieds?tid=78995
```

```
Baseline file located at /var/local/testing/forge/baseline/83  
output from today located at /var/local/testing/forge/current/2003January20-  
trunk/83
```

Then it will have the diff of the two files.

The first line tells you what failed. In the example above the output of today's test did not match the size of the baseline file. The only two options for this line are "File size" or "File data" not matching. This line also contains the number of the test that failed. The second line gives you the URL tested. Next comes the location of the baseline file used to compare with today's output. The final line contains the location of today's output. You can run `diff` on the two files to see what caused the test to fail.