

Debugging AST

Contributed by Administrator
Wednesday, 14 June 2006
Last Updated Wednesday, 14 June 2006

Description of how to run AST in a debugger such as gdb. Can be used to look for bugs in the code or find out what causes "Internal Server Error".

Tools Required

First off you need a debugger, this guide will describe how to use gdb. You may want to use a graphical debugger such as KDbg. The second thing you need is an ast executable that you compiled. You cannot debug the ast executable from binary distribution. You also need to make sure that the debug information is not stripped. If you are following the installation guide, you should skip the strip ast step. Stripping the executable makes it smaller and faster, but removes debugging information.

What to debug?

The first thing you need to do is decide what you want to debug. If you have written some code that is not working, a good way to start is by writing a test case. If you want to debug a scheduled job, you need to find out the name of the job. You can find it in the .cc file for the job. If you are debugging a web page, you need to find the process name for the web page. You can usually find this in the URL, it will say process=.

Command Line Arguments

AST has many command line arguments which are used for debugging. You need to determine which arguments you need to use. For a test case, you need to do:

```
./ast --test
```

If you are debugging a scheduled job, you can use the --run-job argument to invoke it. For example, if you are debugging the alerts job, you might do:

```
./ast --run-job alerts
```

If you are debugging a web page, there are three options of interest to you. --run-process specifies which process to run. --run-as specifies which user to the run job as, and --parameters specifies additional parameters. You should put the query string into --parameters. For example, if you find that the worksheet crashes in view 2, you might do:

```
./ast --run-process worksheet --run-as admin --parameters "process=worksheet&id=2"
```

When you run this command, it will output HTML to the console.

Running AST in gdb

To run AST in gdb, cd into your installation directory and try gdb ast. Now you can type run followed by your arguments. You can use the break command to set a break point (must be done before running). For example, if you want to break in worksheetview.cc at line 321 you might do:

```
# gdb ast
gdb> break worksheetview.cc:321
gdb> run --run-process worksheet --run-as admin --parameters "process=worksheet&id=2"
```

If you want to get a stack trace at the place of a crash, you can type bt. To switch frames in the stack trace type frame 3 (or whatever frame you want to switch to). Then you can print out variables by typing print then the variable.

If you have any questions, please contact me or send an email to the developers mailing list.