

Harmony JDB Command Line Tool

Harmony JDB Command Line Tool

_ Version 0.01 _

1. Motivation & main goal

Currently Harmony do not have a complete and RI compatible implementation of JDB command line tool. The main goal of the project is to implement the JDB command line debugger tool in Java using the Eclipse JDI library. The implementation should be compatible with the Sun JDB command line tool and could be extended with other features - such as tab completion of commands, JSR45 (stratum support), etc. - to add value to the Harmony implementation.

2. Design

Since the JDB implementation is supposed to be compatible enough with RI (Sun JDK), I don't have to specify the commands and function it supports. However, as I will mention in the challenges part, a detailed specification of JDB will be very helpful. I'm planning to create and maintain such an open document during the development of JDB.

For the code framework, the most important point I have to mention is I use enum to implement commands, define match and run methods in the enum class and override them in the enum instance if necessary. I believe the design makes the code structure clear and the only shortage is the long code.

I've divided the commands that jdb supports into two categories:

1. execution control: those commands that control the running of debuggee, such as next, cont, run, stop, etc.
2. information display: those commands that display informations of debuggee when it's suspended, such as threads, print, dump, etc.

and I suggest that developers should concentrate on the first category first because it's more important and related to the code structure.

For those developers who are not familiar with JDI, I'd like to provide a metaphor to describe how to design debugger with JDI: it's not the debugger that drives the program to run (debuggee) or stop; the program just run by itself, if the debugger wants it to stop, set up a barrier (any event request that will cause the program to be blocked) to stop it; if you want it to continue, jump over the barrier (handle the corresponding event) and go on.

3. Challenges

There are several challenges in the implementation of Harmony JDB. For developers who want to continue the work, it's better to take a look at this part and have a serious consideration of how to overcome the difficulties.

*Besides the simple JDB help information of RI, there's no detailed documentation describing and specifying JDB behavior and commands, which makes high compatibility not easy to achieve for developers who are not very familiar with JDB usage.

*JDB supports not only variable value display but also Java expression value, which raise the problem of how to calculate a Java expression in JDB. Regis (Regis Xu) suggested that we need to parse the expression string to find out which part should be calculated at debuggee side, and which can be done at jdb side.

*Matching, parsing and error handling of JDB commands might be easily implemented with straightforward simple algorithm but difficult (at least for me) to find a perfect way. In my implementation, I split the command with spaces and match them by the prefix. For commands with just the same prefix, distinguish them by the following arguments. This makes it easier to match commands and parse them, but fails to handle errors in a uniform way. An alternative way is regular expression matching, which will still need to parse the command. I'm considering whether it's a better way.

4. Achievement so far

The project started as a project of Google Summer of Code 2010. By the time when gsoc 2010 ends, the project has a code base which have a framework or structure of how commands are parsed, matched and executed in jdb, but do not have all commands implemented and there are still some commands do not work well. More, the jdb only support jdb -attach [port] launching. Don't worry, I'll go on with it.

5. Future work

*First of all, implement all of the commands and fix the bugs.

*If time permits, create and maintain a document that specifies jdb.

*Add enhancement to the jdb implementation.

6. Contact info

If you have more questions or have any suggestion, please feel free to contact me daniel.gong.fudan@gmail.com or mail to dev@harmony.apache.org