

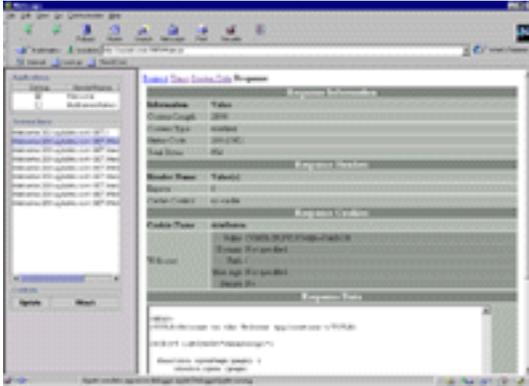
Enhydra 5.1 Debugger Overview

Table of Contents

- 1. Introduction.....
- 2. Installing the Debugger
- 3. Using the Debugger
- 4. Non-Enhydra Servlets

Chapter 1. Introduction

The Enhydra Debugger allows you to debug your Enhydra application that is running in a server other than the Enhydra Multiserver. The Multiserver has its own debugging built in, so you do not have to run the debugger in it, although this debugger incorporates an applet with an improved user interface. The debugger also has the capability to debug third-party (non-Enhydra) servlets.



Chapter 2. Installing the Debugger

The debugger is a servlet, so your server will need to support servlets. In Apache, for example, you should add the JServ servlet runner module. Add a servlet to your server (see the documentation for your server for instructions on how to do this). You will need to supply the following three pieces of information:

- Classpath: `debugger.jar` and `enhydra.jar` (both in `/lib` in your Enhydra installation).
- Class name: `org.enhydra.Servlet`
- Initial parameter: `"ConfFile=/...../apps/debugger.conf"` (replace with the full path of your Enhydra installation, for example `/usr/local/enhydra`).

Adding a servlet is very server-specific, please see your server's documentation.

Chapter 3. Using the Debugger

After you have added the servlet to your server, direct your Web browser to the servlet. The URL you need to use depends on which server you are using and what name you gave the servlet. The main page of the debugger contains instructions on its use.

Chapter 4. Non-Enhydra Servlets

Full instructions are available by running the debugger, going to the main page, and then clicking on the link "flow-through debugging".

In short, suppose the path you use to get to the debugger is "http://host/servlet/debug" (this will depend on your server and how you installed the debugger). Furthermore, suppose the servlet you wish to debug is named "badBoy". Then, the URL to connect to is "http://host/servlet/debug/badBoy/".

The servlet being debugged will be told its servlet path is now "/servlet/debug/badBoy", rather than the usual "/servlet/badBoy". Any absolute URLs generated by the servlet must use this new servlet path (see the method `getServletPath()` in the servlet API) !! That is, you should construct absolute paths using `getServletPath()`, rather than assuming the fixed string "/servlet/badBoy". This is something you should be doing anyway to be server-independent. If you do not construct the absolute URLs correctly then you will only be able to debug your servlet one page at a time: you will not be able to follow links. Relative links should work fine with no changes.