Enhydra Enterprise Server debugging

Table of Contents

Chapter 1. Introduction

For server debugging operations Enhydra uses capabilities and functions of Enhydra Tomcat's and Jetty's debugging system. The functionality of Enhydra Tomcat and Jetty debugging classes is full supported and can therefore be configured in the same way the standard Tomcat or Jetty server is configured.

Chapter 2. Configuring Tomcat

Debug functionality is configured in the server.xml configuration file in the /<enhydra_root>multiserver/conf directory.

For debugging, Enhydra Tomcat uses a system of valves. For the various types of valves and their functionality, check out Tomcat's Valve site [http://localhost:9000/tomcat-docs/config/valve.html] (if this link doesn't work properly checkout *NOTE* below).

NOTE: We are assuming that Enhydra Enterprise is installed, and running, with default connection port setting (9000).

In the *server.xml* file a Valve object type is used to define an *AccessLogValve* that logs all calls to a webserver. The AccessLogValve creates log files in the same format as those created by standard web servers. An example of configuring debugging:

```
<Valve className="org.apache.catalina.valves.AccessLogValve"
directory="logs"  prefix="localhost_access_log." suffix=".txt"
 pattern="common" resolveHosts="false" />
```

- className: Java class name of the implementation to use. This MUST be set to org.apache.catalina.valves.AccessLogValve.
- directory: Absolute or relative pathname of a directory in which log files created by this valve will be placed.
- prefix: The prefix added to the start of each log file's name. If not specified, the default value is "access_log.". To specify no prefix, use a zero-length string.
- suffix: The suffix added to the end of each log file's name. If not specified, the default value is "". To specify no suffix, use a zero-length string.
- pattern: A formatting layout identifying the various information fields from the request and response to be logged, or the word common or combined to select a standard format.
- resolveHosts: Set to true to convert the IP address of the remote host into the corresponding host name via a DNS lookup.

For a detailed explanation of these and other parameters' definition, please check the Documentation [http://localhost:9000/tomcat-docs/config/valve.html] (if this link doesn't work properly checkout *NOTE* below).

NOTE: We are assuming that Enhydra Enterprise is installed, and running, with default connection port setting (9000).

Chapter 3. Configuring Jetty

Debug functionality is configured in the jetty.xml configuration file in the enhydra/conf folder.

Enhydra Jetty uses its NCSARequestLog to handle request logs. NCSARequestLog can be configured in a fairly straight forward way. An example of possibilities:

```
<Configure class="org.mortbay.jetty.Server">
  ...
  <Set name="RequestLog">
    <New class="org.mortbay.http.NCSARequestLog">
      <Set name="filename">./logs/yyyy_mm_dd.request.log</Set>
      <Set name="buffered">false</Set>
      <Set name="retainDays">90</Set>
      <Set name="append">true</Set>
      <Set name="extended">false</Set>
      <Set name="logTimeZone">GMT</Set>
      <Set name="ignorePaths">
        <Array type="String">
          <Item>/images/*</Item>
          <Item>*.css</Item>
        </Array>
      </Set>
    </New>
  </Set>
  ...
</Configure>
```

More about NCSARequestLog and other Jetty classes at Jetty Homepage [http://jetty.mortbay.org/javadoc/index.html]

Chapter 4. Using debugger

There is no special procedure for using the debugger. i.e. Start your server, point to its location in your favorite web browser, start up some applications, and off you go! The debug information with selected parameters will be written to the log file selected.

NOTE: We developed enhydra Debugger application that may help yuo during application debugging process. Please, check out Enhydra Debugger [../developer/debugger.xml] application document for further information!