BEA WebLogic Server ^{R} Servlet Container Debug Enhancement

Version: 10.3 Tech Preview

Document Date: October 2007

Table of Contents

Overview of Servlet Container Debug Enhancement 3
Disabling Access Logging 3
    Purpose 3
    Usage 3
Debugging Specific Session 3
    Purpose 4
    Usage 4
Tracking Request Handle Footprint 4
    Purpose 4
    Usage 4

Overview of Servlet Container Debug Enhancement

The following sections provide information on debug support enhancements made in the WebLogic Server (WLS) servlet container:

- Disabling Access Logging
- Debugging specific session
- Tracking request handle footprint

Disabling Access Logging

The following sections provide information on how to disable access logging.

Purpose

Logging access can be expensive. In cases where access logging is not required, you can improve server performance by disabling access logging.

Usage

A new optional property “disable-access-logging” is introduced into “container-descriptor” in weblogic.xml to indicate if access logging is disabled.

- If the property is defined with a value “true”, accesses to the underlying web application are not logged to access log file
- If the property is defined with a value “false” explicitly, accesses are logged.
- If the property is not defined, accesses are logged when the web application is a normal application. It is not logged in case where the web application is a WLS internal application.

Note that the property is at the web application level. The property defined in a web application does not affect other web applications. This property works under both development mode and production mode.

The following is an example that disables access logging:

weblogic.xml file:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
  <weblogic-web-app xmlns="http://www.bea.com/ns/weblogic/90">
    <container-descriptor>
      <disable-access-logging>true</disable-access-logging>
    </container-descriptor>
</weblogic-web-app>
```

Debugging Specific Session

The following sections provide information on how to debug specific sessions.

Purpose

Tracking session change is very helpful for developers, especially in case of replicated sessions. Although you can utilize HttpSessionAttributeListener to track session changes, it is web application level. Developers need a finer grain debug option to track session change during a specific request.

Usage

A request flag named “wl_debug_session” is introduced to log the changes of current request in the current session. If the flag is used, the container logs the modifications of the underlying session into server log.

The following is an example of a request that uses the wl_debug_session parameter:

http://localhost/foocontext/foo?name=abc&wl_debug_session

Note that the debug option is in the request scope and is available only under development mode. In production mode, the parameter is treated as a normal request parameter.

Tracking Request Handle Footprint

The following sections provide information on tracking a request handle footprint.

Purpose

Tracking a request handle footprint is very helpful in development mode. When debugging an application, you need to know many pieces of information. This includes such information as: what request is received, how it is dispatched, what session is bound to it, when servlet is invoked, and what response is sent. Finally, when a ServletException occurs, you need a way to link the exception to corresponding request to find the root cause of the error.

Usage

For this release, the WLS servlet container provides more detailed log messages during request handling to better describe the each milestone in a request flow. No additional configuration changes are required other than enabling the HttpDebug logger.

You can then find the footprint of a request handle in the server log. Once in production mode, you should disable HttpDebug logger to maximize server performance.