Java CAPS 5.1 Field Notes

Command line management of Sun SeeBeyond Integration Server with isadmin

Note 1: Runtime logging management

December 2007

Table of Contents

Introduction

In Java CAPS 5.1 the Sun SeeBeyond Integration Server is a modified version of the Sun Application Server 8.0 Platform Edition. As such a number of things that can be done with that application server can be done with the Sun SeeBeyond Integration Server. Most notably, it can be managed at runtime using the `isadmin` tool much the same way as the Sun Application Server 8.0 PE can be managed using the `asadmin` tool.

In as much as the `isadmin` tool is barely mentioned in Java CAPS documentation the material presented here is gathered through empirical study rather then gleaned from documentation.

Additional material on the Sun Application Server 8.0 Platform Edition can be found on Sun's documentation site and elsewhere:
http://docs.sun.com/app/docs/coll/ApplicationServer8_04q2
http://java.sun.com/j2ee/1.4/docs/tutorial/doc/Overview8.html

This note discusses how the `isadmin` tool can be used to manage runtime logging of the Sun SeeBeyond Integration Server. This note applies to the Sun SeeBeyond Integration Server included in Java CAPS 5.1.

Determining logging configuration

This command will list all log-service configuration properties, for the Integration Server running on port 20000, and their values.

```
isadmin get -p 20000 server.log-service.*
server.log-service.alarms = false
server.log-service.file =
C:/JCAPS513/logicalhost/is/domains/wssec/logs/server.log
```

```
server.log-service.log-filter = <null>
server.log-service.log-handler = <null>
server.log-service.log-rotation-limit-in-bytes = 10000000
server.log-service.log-to-console = false
server.log-service.property.max-msg-per-second = 0
server.log-service.property.parole-in-millis = 500
server.log-service.property.penalty-in-millis = 250
server.log-service.property.print-dup-stack-trace = false
server.log-service.property.print-sysout-warning = false
server.log-service.property.recycle-size = 10
server.log-service.use-system-logging = false
```

For explanation of the properties see http://docs.sun.com/source/817-6093/crdomxml.html#wp1006045

In a Unix-like environment, or in a Windows environment with the `grep` command available, one can issue the following `isadmin` command to obtain a list of all configuration properties pertaining to the log-service hierarchy and their current values.

```
C:\JCAPS513\logicalhost\is\bin>isadmin get -p 20000 server*.* | grep
log-service
server.log-service.alarms = false
server.log-service.file =
C:/JCAPS513/logicalhost/is/domains/wssec/logs/server.log
server.log-service.log-filter = <null>
server.log-service.log-handler = <null>
server.log-service.log-rotation-limit-in-bytes = 10000000
server.log-service.log-to-console = false
server.log-service.module-log-levels.admin = INFO
server.log-service.module-log-levels.classloader = INFO
server.log-service.module-log-levels.cmp = INFO
server.log-service.module-log-levels.cmp-container = INFO
server.log-service.module-log-levels.configuration = INFO
server.log-service.module-log-levels.connector = INFO
server.log-service.module-log-levels.corba = INFO
server.log-service.module-log-levels.deployment = INFO
server.log-service.module-log-levels.ejb-container = INFO
server.log-service.module-log-levels.javamail = INFO
server.log-service.module-log-levels.jaxr = INFO
server.log-service.module-log-levels.jaxrpc = INFO
server.log-service.module-log-levels.jdo = INFO
server.log-service.module-log-levels.jms = INFO
server.log-service.module-log-levels.jta = INFO
server.log-service.module-log-levels.jts = INFO
server.log-service.module-log-levels.mdb-container = INFO
server.log-service.module-log-levels.naming = INFO
server.log-service.module-log-levels.property.com = FINEST
server.log-service.module-log-levels.resource-adapter = INFO
server.log-service.module-log-levels.root = INFO
server.log-service.module-log-levels.saaj = INFO
server.log-service.module-log-levels.security = INFO
server.log-service.module-log-levels.server = INFO
server.log-service.module-log-levels.verifier = INFO
server.log-service.module-log-levels.web-container = INFO
server.log-service.property.max-msg-per-second = 0
server.log-service.property.parole-in-millis = 500
server.log-service.property.penalty-in-millis = 250
server.log-service.property.print-dup-stack-trace = true
```

```
server.log-service.property.print-sysout-warning = false
server.log-service.property.recycle-size = 10
server.log-service.use-system-logging = false
```

Note that `server.log-service.module-log-levels` component lists all module log levels but not the log levels for specific logging categories. Given a domain.xml with the following content fragment:

```
<log-service
  alarms="false"
  file="C:/JCAPS513/logicalhost/is/domains/wssec/logs/server.log"
  log-rotation-limit-in-bytes="10000000"
  log-to-console="false"
  use-system-logging="false"
  >
   <module-log-levels
          admin="INFO"
          classloader="INFO"
          cmp="INFO"
          .
          .
          .
          verifier="INFO"
          web-container="INFO"
          >
             <property name="com.sun.xml.wss.logging.impl.crypto" value="INFO"/>
             <property name="org.apache.xalan" value="INFO"/>
             <property name="STC.eWay.ExConfigSvc" value="INFO"/>
             <property name="com.sun.xml.wss" value="INFO"/>
             <property name="javax.xml.soap" value="FINEST"/>
             <property name="STC.eWay.framework.management" value="INFO"/>
             <property name="com.sun.identity" value="FINEST"/>
             .
             .
             .
```

one can readily relate specific elements to the `server.log-service` components:

```
server.log-service.alarms = false
server.log-service.file =
C:/JCAPS513/logicalhost/is/domains/wssec/logs/server.log
server.log-service.log-filter = <null>
server.log-service.log-handler = <null>
server.log-service.log-rotation-limit-in-bytes = 10000000
server.log-service.log-to-console = false
server.log-service.module-log-levels.admin = INFO
server.log-service.module-log-levels.classloader = INFO
server.log-service.module-log-levels.cmp = INFO
.
.
.
server.log-service.module-log-levels.verifier = INFO
server.log-service.module-log-levels.web-container = INFO
```

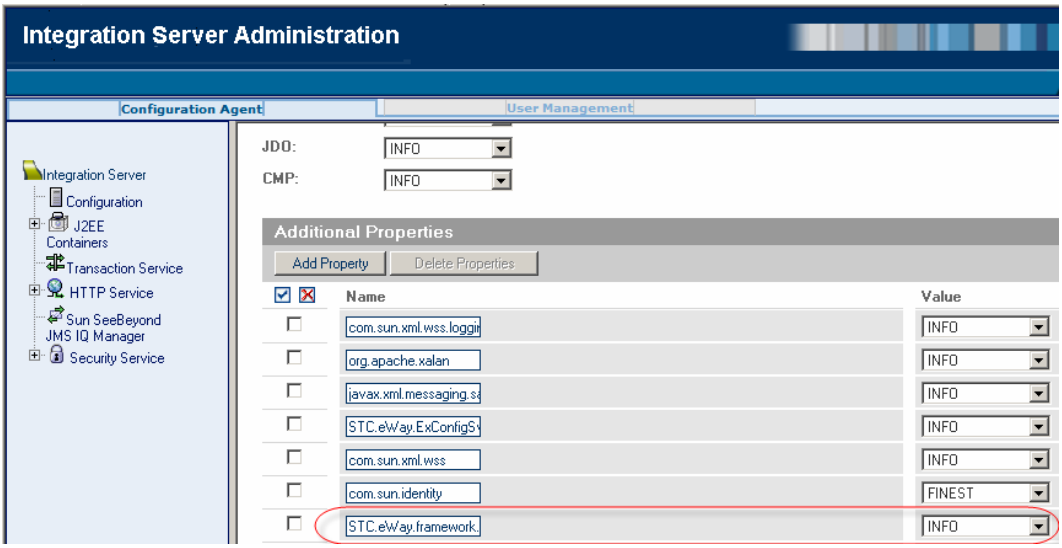
however none of the `property` elements below module-log-levels, present in the domain.xml, are listed using the `isadmin get` command. I have not found a way to list these elements using the `asadmin` or the `isadmin` tools.

Changing log levels for specific logging categories

Inspection of the domain.xml for a Java CAPS domain may reveal a series of properties with names of logging categories under the module-log-levels element, for example:

```
<module-log-levels
       admin="INFO"
       classloader="INFO"
       cmp="INFO"
       .
       .
       .
       verifier="INFO"
       web-container="INFO"
       >
          <property name="com.sun.xml.wss.logging.impl.crypto" value="INFO"/>
          <property name="org.apache.xalan" value="INFO"/>
          <property name="STC.eWay.ExConfigSvc" value="INFO"/>
          <property name="com.sun.xml.wss" value="INFO"/>
          <property name="javax.xml.soap" value="FINEST"/>
          <property name="STC.eWay.framework.management" value="INFO"/>
          .
          .
          .
```

These properties would have typically been set through the Integration Server Administration console, logging Tab, log-levels sub-tab.



To display the values of a specific property, STC.eWay.framework.management, using the `isadmin` tool one would enter a command like similar to the following:

```
C:\JCAPS513\logicalhost\is\bin>isadmin get -p 20000 server.log-service.module-log-levels.property.STC\.eWay\.framework\.management
server.log-service.module-log-levels.property.STC.eWay.framework.management = INFO
```

Note that all dots “.” in the logging category STC.eWay.framework.management have been escaped with the backslash “\”.

To change the value of the logging category one would issue the following command, where log level to set is one of OFF, SEVER, WARNING, INFO, CONFIG, FINE, FINER, FINEST:

```
C:\JCAPS513\logicalhost\is\bin>isadmin set -p 20000 server.log-service.module-log-levels.property.STC\.eWay\.framework\.management=F
```

```
INE
server.log-service.module-log-levels.property.STC.eWay.framework.management = FINE
```

The command executes and displays the current logging level for the category.

Note that specifying partial category name for the `get` command does not return a partial category but results in an error, for example:

```
C:\JCAPS513\logicalhost\is\bin>isadmin get -p 20000 server.log-service.module-log-
levels.property.STC\.eWay\.framework
STC.eWay.framework
CLI137 Command get failed.
```

Issuing a set command with a category name not already in the domain.xml causes that category property element to be created and set in the domain.xml:

```
C:\JCAPS513\logicalhost\is\bin>isadmin set -p 20000 server.log-service.module-log-
levels.property.STC\.eWay\.framework=CONFIG
server.log-service.module-log-levels.property.STC.eWay.framework = CONFIG

C:\JCAPS513\logicalhost\is\bin>isadmin get -p 20000 server.log-service.module-log-
levels.property.STC\.eWay\.framework
server.log-service.module-log-levels.property.STC.eWay.framework = CONFIG
```

Successful command execution sets the ERRORLEVEL to 0 on Windows.

Unsuccessful command execution sets the ERRORLEVEL to 1 in Windows.

In Unix-like environments command execution status is set at the shell level.

Command execution status can be used in scripts/batch files to control script/batch file execution.

More examples

Get the value of the log level for the logging category javax.xml.messaging.saaj

```
isadmin get -p 20000 ^
server.log-service.module-log-levels.property.javax\.xml\.messaging\.saaj
```

Note that the category `javax.xml.messaging.saaj`, being already a dotted name, is escaped as `javax\.xml\.messaging\.saaj`

This command produces the following:

```
server.log-service.module-log-levels.property.javax.xml.messaging.saaj = FINEST
```

To change this logging category logging level to INFO one would use the set command:

```
isadmin set -p 20000 ^
server.log-service.module-log-levels.property.javax\.xml\.messaging\.saaj=INFO
```

which would result in the following output:

```
server.log-service.module-log-levels.property.javax.xml.messaging.saaj = INFO
```