

## SEEK - Bug #2422

### Changing Logging Options Prevent Metadata Downloads

04/20/2006 11:37 AM - David Sledge

<b>Status:</b>	New	<b>Start date:</b>	04/20/2006
<b>Priority:</b>	Low	<b>Due date:</b>	
<b>Assignee:</b>	Matt Jones	<b>% Done:</b>	0%
<b>Category:</b>	ecogrid	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	Unspecified	<b>Spent time:</b>	0.00 hour
<b>Bugzilla-Id:</b>	2422		
<b>Description</b>			
<p>In the file log4j.properties in kepler's base directory, changing the line that reads:</p> <pre>log4j.rootLogger=WARN, CONSOLE, R</pre> <p>to:</p> <pre>log4j.rootLogger=DEBUG, CONSOLE, R</pre> <p>prevents Metadata from being downloaded from Metacat. With the ecogrid webapp--which uses the same API that kepler uses to access data on the ecogrid--the lack of a log4j.properties file prevented the metadata from being downloaded.</p> <p>In both cases, org.ecoinformatics.ecogrid.queryservice.stub.EcoGridQueryLevelOneServiceStub.get() threw a ClassCastException.</p> <p>Kepler:</p> <pre>java.lang.ClassCastException: java.lang.String at org.ecoinformatics.ecogrid.queryservice.stub.EcoGridQueryLevelOneServiceStub.get(EcoGridQueryLevelOneServiceStub.java:310) at org.ecoinformatics.ecogrid.queryservice.EcogridGetToStreamClient.get(EcogridGetToStreamClient.java:81) ...</pre> <p>Webapp:</p> <pre>java.lang.ClassCastException: java.lang.String at org.ecoinformatics.ecogrid.queryservice.stub.EcoGridQueryLevelOneServiceStub.get(Unknown Source) at org.ecoinformatics.ecogrid.queryservice.EcogridGetToStreamClient.get(Unknown Source) ...</pre>			

#### History

##### #1 - 04/21/2006 11:02 AM - David Sledge

I'm changing the severity of this bug, due to that I'm now getting the exception again in the webapp, but I haven't changed the logging settings in log4j.properties since the file's inclusion to the webapp.

##### #2 - 04/25/2006 03:47 PM - David Sledge

As best as I can tell it's a bug in the axis 1.3 library. I removed all the logging code from EcogridGetToStreamClient in the ecogrid API, and I put together some simple test code:

```
import java.io.ByteArrayOutputStream;
import java.net.URL;

import org.ecoinformatics.ecogrid.queryservice.EcogridGetToStreamClient;

public class test {
    public static void main(String[] args) {
        try {
            URL endPoint = new URL();
            EcogridGetToStreamClient client = new EcogridGetToStreamClient(endPoint);
            ByteArrayOutputStream baos = new ByteArrayOutputStream();

            client.get("knb.182.1", baos);
        }
    }
}
```

```
        System.out.println(baos.toString());
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}
```

The results were predictably the same as in kepler.

The workaround for this in a servlet container is to make sure the Commons Logging and log4j jar files are included in the webapp's WEB-INF/lib directory (versus \$CATALINA\_HOME/shared/lib or \$CATALINA\_HOME/common/lib), the files commons.properties and log4j.properties are put in WEB-INF/classes, and that the logging is set to INFO or something coarser.

### **#3 - 03/27/2013 02:20 PM - Redmine Admin**

Original Bugzilla ID was 2422