LTERgrid System Configuration

Updated: September 15, 2005

grid.matrix.msu.edu

Incoming Ports

Firewalls, iptables, etc., need to allow incoming connections on the following TCP/IP ports:

Port	Description
22	Secure Shell
2811	GridFTP Control Channel
2119	GRAM
50000-51000	Controllable Ephemeral Port Range for GRAM/GridFTP Data Channels

/etc/xinet.d/grid-ftp

```
env += GLOBUS_TCP_PORT_RANGE=50000,51000
```

roadrunner.lternet.edu

Incoming Ports

Firewalls, iptables, etc., need to allow incoming connections on the following TCP/IP ports:

Port	*Description*
22	Secure Shell
514	syslog-ng
80	Apache
443	Apache SSL
5432	Postgres Database
8080	Tomcat (pilotapp)
8443	Tomcat SSL (metacat)
8888	Tomcat (metacat)
7512	MyProxy Credential Repository
50000-51000	GRAM/GridFTP Callback Connections

iptables

```
iptables -t nat -I PREROUTING -p tcp -i eth0 --dport 80 -j REDIRECT --to-port 8080
```

/etc/rc.d/rc.local

For syslog-ng add,

```
nohup /usr/local/etc/syslog-ng/pgsql_syslog.sh &
```

/var/lib/pgsql/data/pg_hba.conf

Allow remote access to database

Tomcat

Changes to Tomcat to support an HTTPS+GSI client

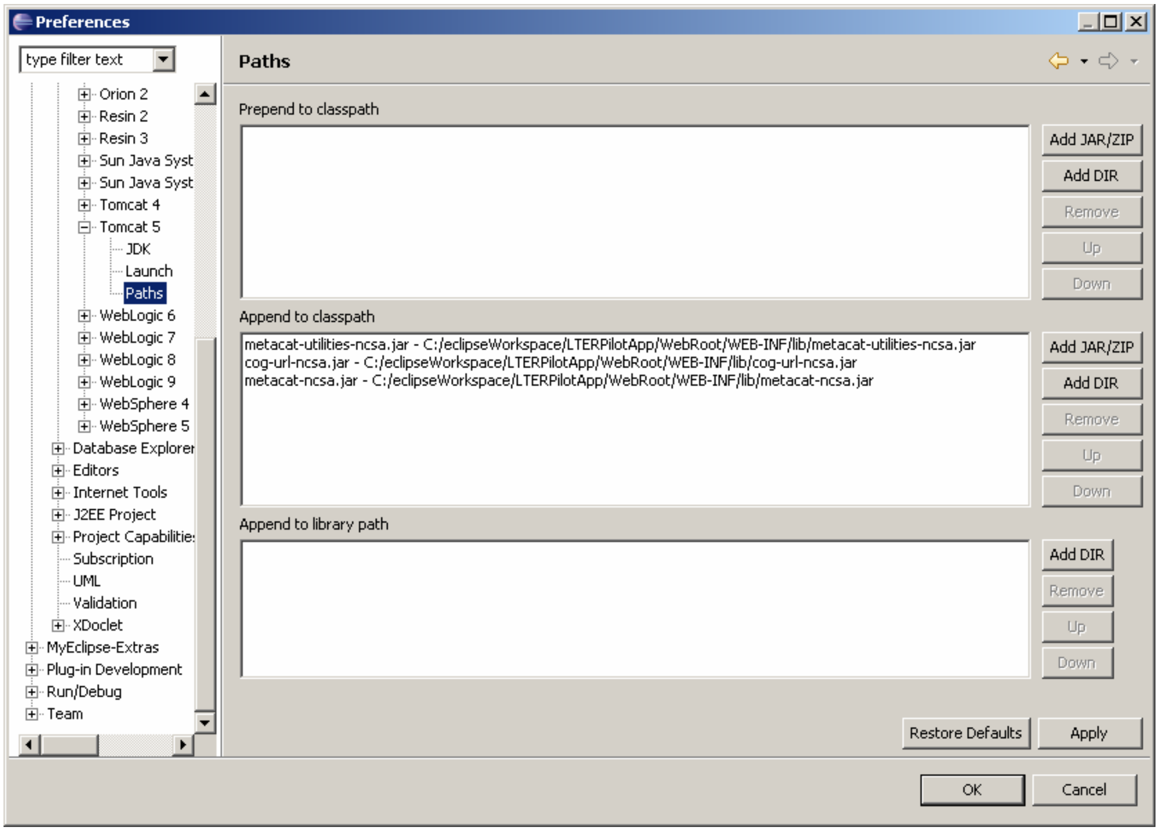
In order to support the LTER pilot web app communicating with a Metacat server via HTTPS+GSI, we need to make two changes to Tomcat:

1. Add JARs to Tomcat’s root classpath, in tomcat/bin/catalina.sh:
    a. metacat-ncsa.jar
    b. metacat-utilities-ncsa.jar
    c. cog-url-ncsa.jar

```
[root@roadrunner bin]# diff catalina.sh catalina.sh~

<CLASSPATH="$CLASSPATH":"$CATALINA_HOME"/bin/bootstrap.jar:"$CATALINA_HOME"/bin/c
ommons-logging-api.jar:"$CATALINA_HOME"/common/lib/metacat-
ncsa.jar:"$CATALINA_HOME"/common/lib/metacat-utilities-
ncsa.jar:"$CATALINA_HOME/common/lib/cog-url-ncsa.jar"
---
>CLASSPATH="$CLASSPATH":"$CATALINA_HOME"/bin/bootstrap.jar:"$CATALINA_HOME"/bin/c
ommons-logging-api.jar
```

For the eclipse environment, specify these jars in Window → Preferences → MyEclipse → Application Servers → Tomcat 5 → Paths:



Append to classpath

```
metacat-utilities-ncsa.jar - C:/eclipseWorkspace/LTERPilotApp/WebRoot/WEB-INF/lib/metacat-utilities-ncsa.jar
cog-url-ncsa.jar - C:/eclipseWorkspace/LTERPilotApp/WebRoot/WEB-INF/lib/cog-url-ncsa.jar
metacat-ncsa.jar - C:/eclipseWorkspace/LTERPilotApp/WebRoot/WEB-INF/lib/metacat-ncsa.jar
```

Why and How

Java has several mechanisms for assigning protocol handlers, but we are restricted in Tomcat to specifying a system property from which Java can derive protocol handler class names. The classes that we specify this way have to be in Tomcat's root classpath – that is, the classpath that Tomcat starts up with, before it dynamically loads JARs from common/lib or from application-specific lib dirs.

To add to Tomcat's root classpath, change its startup script, catalina.sh/catalina.bat. Change the line where the variable CLASSPATH is first defined (it initially includes bootstrap.jar and commons-logging-api.jar) to include the new JARs.

The protocol that we use for HTTPS+GSI is "httpg". Really, it could be anything (metacatgsi, httpgsi, etc.), as long as Tomcat isn't already familiar with it, because we want to provide our own special HTTPS+GSI protocol handler. Unfortunately, if Java has already assigned a handler to a particular protocol, there is no way to *un*-assign that handler. In other environments, you could preempt that initial assignment and, for example, supply a custom HTTPS handler that understands HTTPS+GSI.

Note on HTTPG: Globus Toolkit version 3 (GT3) uses this protocol name as well, but we can’t get their “httpg” handler to work – only the “https” handler. So we’ve wrapped the GT3 HTTPS handler as a Metacat-specific HTTPG handler. That’s something to revisit – why does the GT3 HTTPG handler not work for us whereas the GT3 HTTPS handler does?