Adding Grid Security

The information and links on this page are no longer being updated and are provided for reference purposes only.

Contents of this Page

Purpose

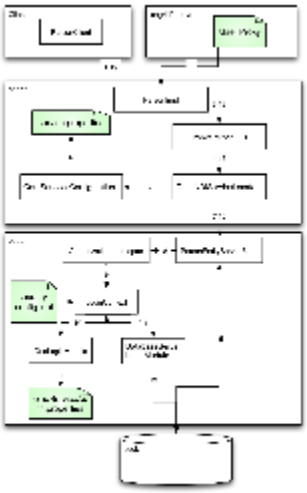
To provide a lightweight guide for other CBIIT applications (eg, caArray) to secure their own grid services. This implementation combines caGrid authentication and local CSM authorization.

Technical Details

Overview of Grid Security Workflow

Workflow Steps

```
1. Client request sent
2. Grid User's proxy is obtained from the default location on the file system and verified.
3. PersonImpl getById is invoked, calling InvokePersonEJB
4. InvokePersonEJB obtains the caller identity from the proxy and instantiates a GridJNDIServiceLocator
5. GridJNDIServiceLocator obtains default Grid Service credentials from service.properties via
CoreServicesConfiguration and creates the InitialContext.
6. CoreServicesConfiguration reads the property file
7. InvokePersonEJB calls the getPerson method via the po/PersonEntityServiceBean/remote.
8. AuthorizationInterceptor is invoked
9. LoginContext, configured by security-config.xml, starts authenticationa and authorization
10. GridLoginModule is invoked. Details provided above.
11. GridLoginModule obtains the expected principal/encrypted password from GridServiceAuth.properties
12. DatabaseServerLoginModule verifies login success
13. DatabaseServerLoginModule queries podb CSM tables to obtain the Roles associated with our Grid
("gridClient" is expected)
14. Role is set in Login sharedState
15. PersonEntityServiceBean verifies required "client or gridClient" role is present and queries the DB to
obtain the person record.
16-18. Person data is returned to the Grid User.
```



Download a PDF version of the diagram COPPASecurity_508_compliant.pdf

Assumptions

- JBoss 4.0.5
- JAAS
- Remote EJBs for business application integration
- caGrid 1.3
- Using BDA for JBoss container configuration of secure services
- Using Common Security Module (CSM)

Changes to Business Application

Assumptions

- JBoss 4.0.5
- JAAS
- Existing Secured Remote EJBs
- Using Common Security Module (CSM)

1. Add CommonsGridLoginModule to JAAS login module (security-config.xml)
   - ⚠️ requires `nci-commons-core` version 1.2.4 or greater see http://maven.5amsolutions.com/archiva/browse/com.fiveamsolutions/nci-commons-core
   - ⚠️ requires `jbosssx.jar` as runtime dependency to handle decryption of encrypted pre-shared key within `CommonsGridLoginModule` class. Typically included with JBoss by default, please verify.

Add to JAAS Login Module (security-config.xml)

```
<login-module code="com.fiveamsolutions.nci.commons.authentication.CommonsGridLoginModule" flag="
optional">
    <module-option name="gridServicePrincipal">${gridServicePrincipal}</module-option>
    <module-option name="gridServiceCredential">${gridServiceCredential}</module-option>
    <module-option name="gridServicePrincipalSeparator">||</module-option>
</login-module>
```

   - Define gridServicePrincipal & gridServiceCredential properties within appropriate properties file so that the login module configuration file is properly configured as a part of the build and deployment process for your application

Example snippet to add Maven2 properties

```
<gridServicePrincipal>Gr1DU5er</gridServicePrincipal>
        <gridServiceCredential>ltHZmZ1rqYq8j2uyHEABIQ==</gridServiceCredential>
```

ℹ️ The unencrypted value for `ltHZmZ1rqYq8j2uyHEABIQ==` is `Pa44Wurd`

2. Introduce a new grid service instance CSM Group
   ⚠️ Update the application name `'po'` to your application's name

Sample SQL for Postgres to define a new CSM Group

```
INSERT INTO CSM_GROUP (GROUP_NAME, GROUP_DESC, APPLICATION_ID)
VALUES ('gridClient', 'Grid Service Invocation Group', (select application_id from csm_application where
application_name = 'po'));
```

3. Update @Remote EJBs endpoints to allow the new CSM Group using the @RolesAllowed annotation

Example with only grid access

```
@RolesAllowed("gridClient")
public void myRemoteEndpointMethod() { ... }
```

Example granting both grid and web clients access

```
@RolesAllowed({"webClient","gridClient"})
public void myRemoteEndpointMethod() { ... }
```

Changes to Grid Service(s)

Assumptions

- caGrid 1.3
- JBoss 4.0.5 Grid Service Deployment
- Using BDA for JBoss container configuration of secure services
- Secured Remote EJBs for Business Application integration

1. Alter Service Context(s) within Introduce
   - Modify each service context accordingly to add security
     a. Highlight Service Context, click Modify Service button
     b. Under Information Page, User Resource Framework Options section, **check *Secure***
     c. Under Security Page (tab/button at top of dialogue), **choose Custom**
     d. Then under Secure Communication tab, **check *Transport Layer Security*** , choose **Privacy** for *Communication Method*
        ℹ️ Specifying *Transport Layer Security* (TLS) enables encryption
     e. Then under Authorization tab, select **No** for *Client should connect anonymously* AND select **Enforce Authentication** for *Authorization Mechanism*
        ℹ️ These settings force the user to authenticate with the Grid and provide a valid user credential when calling the grid data service
     f. Then under Service Credentials tab, select **System** for *Run As*
   - Add Service Property to your *(Main Service)*context within Introduce,
     a. Select Service Properties tab, input the following values:

Key	Default Value	Description
gridServicePrincipalSeparator	||	The separator used to encord the gridServicePrincipal and grid user's identity when Using the com.fiveamsolutions.nci.commons.authentication.CommonsGridLoginModule

     b. Click *Add* button
   - Ensure the appropriate *Types*are included within your grid service, if not add the types (XSDs) by doing the following:
     a. Import Data Types -> caDSR; Project: caGrid_Metadata_Models (version 1); Package gov.nih.nci.cagrid.metadata.security
   - Save your changes within Introduce (must be successful)
2. Ensure the Service Property is specified within `service.properties`

```
#service deployment properties
#Wed Nov 04 17:13:39 EST 2009
gridServicePrincipalSeparator=||
```

3. Alter how remote services (eg, EJBs) are authenticated and authorized for each grid service request.
   As an example, create a `GridSecurityJNDIServiceLocator` class to authenticate using both the Grid User's Identity (eg, `/O=caBIG /OU=caGrid/OU=Training/OU=Dorian/CN=coppagridtest` instead of a typical remote service user. In short, you'll base your implementation off of your existing Locator (eg, `JNDIServiceLocator`) and replace existing occurrences with the new `GridSecurityJNDIServiceLocator`.
   ⚠️ Don't forget to update the values for the `java.naming.security.principal` and `java.naming.security.credentials` when using the the new `GridSecurityJNDIServiceLocator`, see example below.

```
<property name="java.naming.security.principal" value="Gr1DU5er" />
<property name="java.naming.security.credentials" value="Pa44Wurd" />
```

See https://ncisvn.nci.nih.gov/svn/coppa/trunk/code/po-grid/src/gov/nih/nci/coppa/po/grid/remote/GridSecurityJNDIServiceLocator.java for full code

Below is an example that demonstrates the *essence* of how to code it up your new `GridSecurityJNDIServiceLocator` class.

About example

`CoreServicesConfiguration` is the `ServiceConfiguration` for our *(Main Service)* context that you previously added a Service Property when updating your services using Introduce.

`GridSecurityJNDIServiceLocator` may not be a singleton (static) within your application as the contained `InitialContext` instance needs to reference the Grid Identity for the incoming request by using `SecurityUtils.getCallerIdentity()`.

While this is recognized as a performance hit, we've yet to figure a better way. If anyone is able to determine a better way, please let the COPPA team know team-po@5amsolutions.com --thanks

Essentials for a GridSecurityJNDIServiceLocator implementation

```
...
    private InitialContext context;
    private static final String JNDI_PRINCIPAL = "java.naming.security.principal";
    private static final String JNDI_CREDENTIALS = "java.naming.security.credentials";

    /**
     * @return a ServiceLocator with the caller's identity
     * @throws Exception if a problem occurs
     */
    public static ServiceLocator newInstance() throws Exception {
        return new GridSecurityJNDIServiceLocator(SecurityUtils.getCallerIdentity());
    }

    /**
     * Get an instance of the service locator. specific to the grid user.
     *
     * @param userIdentity user identity of the grid user
     */
    public GridSecurityJNDIServiceLocator(String userIdentity) {
        try {
            Properties props = new Properties();
            props.load(GridSecurityJNDIServiceLocator.class.getClassLoader().getResourceAsStream("jndi.
properties"));

            // set grid service principal and grid identity as java.naming.security.principal
            CoreServicesConfiguration coreConfiguration = CoreServicesConfiguration.getConfiguration();
            String principal = props.getProperty(JNDI_PRINCIPAL)
                    + coreConfiguration.getGridServicePrincipalSeparator() + userIdentity;
            props.setProperty(JNDI_PRINCIPAL, principal);

            LOG.debug("Properties " + props.toString());

            context = new InitialContext(props);

        } catch (Exception e) {
            LOG.error("Unable to load jndi properties.", e);
            throw new RuntimeException("Unable to load jndi properties.", e);
        }
    }


    private Object lookup(String name) throws NamingException {
        Object object = null;
        int i = 0;
        while (object == null && i < MAX_RETRIES) {
             try {
                 LOG.debug("Performing JNDI Lookup of : " + name);
                 object = context.lookup(name);
             } catch (CommunicationException com) {
                 LOG.warn("Unable to lookup: " + name);
             }
             i++;
        }

        return object;
    }

    /**
     * {@inheritDoc}
     */
    public PersonEntityServiceRemote getPersonService() throws NamingException {
        PersonEntityServiceRemote object = (PersonEntityServiceRemote) lookup("po/PersonEntityServiceBean
/remote");
        return object;
    }
...
```

Next, an example of demonstrating the use of the `GridSecurityJNDIServiceLocator` class

Using GridSecurityJNDIServiceLocator

```
/**
     * {@inheritDoc}
     */
    public PersonDTO getPerson(Ii ii) throws NullifiedEntityException {

        try {
            PersonDTO person = GridSecurityJNDIServiceLocator.newInstance().getPersonService().getPerson
(ii);
            return person;
        } catch (NullifiedEntityException e) {
            throw e;
        } catch (UndeclaredThrowableException e) {
            throw (e);
        } catch (Exception e) {
            throw new InvokeCoppaServiceException(e.toString(), e);
        }
    }
```

Lastly, here are the JNDI Properties

jndi.properties

```
java.naming.factory.initial=${java.naming.factory.initial}
java.naming.provider.url=${java.naming.provider.url}
java.naming.factory.url.pkgs=${java.naming.factory.url.pkgs}
java.naming.security.principal=${java.naming.security.principal}
java.naming.security.credentials=${java.naming.security.credentials}
```

Be sure to filter the values as a part of your build process

```
<property name="java.naming.factory.initial" value="org.jboss.security.jndi.
JndiLoginInitialContextFactory" />
<property name="java.naming.provider.url.host" value="localhost" />
<property name="java.naming.provider.url.port" value="1099" />
<property name="java.naming.provider.url" value="jnp://${java.naming.provider.url.host}:${java.naming.
provider.url.port}" />
<property name="java.naming.factory.url.pkgs" value="org.jboss.naming:org.jnp.interfaces" />
<property name="java.naming.security.principal" value="Gr1DU5er" />
<property name="java.naming.security.credentials" value="Pa44Wurd" />
```

Changes to BDA scripts

This section will likely vary based on many factors and more notably your specific version of BDA and existing deployment configuration steps.

1. Consult How to configure a Secure Grid Listener

Below is a diff of the changes for COPPA-PO BDA Scripts:

Diff of the changes to add grid security for COPPA-PO

```
Index: install.properties
===================================================================
--- install.properties        (revision 7040)
+++ install.properties        (revision 7192)
@@ -118,12 +118,20 @@
 po-grid.jboss.snmp-trapd.port=21362
 po-grid.jboss.web.service.port=28283
```

```
 #used to allow the po-grid to make EJB3 calls via jndi
-pogrid.jndi.principal=ejbclient
-pogrid.jndi.credentials=pass
+pogrid.jndi.principal=Gr1DU5er
+pogrid.jndi.credentials=Pa44Wurd

-pogrid.jboss.external.http.host=localhost
-pogrid.jboss.external.http.port=29280
+pogrid.jboss.external.http.host=${pogrid.grid.external.secure.host}
+pogrid.jboss.external.http.port=${pogrid.grid.secure.port}

+pogrid.grid.secure.enable=true
+pogrid.grid.secure.port=29443
+pogrid.grid.secure.cert.location=${user.home}/.cagrid/certificates/${pogrid.grid.external.secure.host}-cert.pem
+pogrid.grid.secure.key.location=${user.home}/.cagrid/certificates/${pogrid.grid.external.secure.host}-key.pem
+pogrid.grid.external.secure.host=${env.HOSTNAME}
+pogrid.grid.external.secure.port=29443
+
+
 po-grid-legacy.jboss.server.name=pogridlegacy
 po-grid-legacy.jboss.server.jndi.port=21099
 po-grid-legacy.jboss.server.port=29080
@@ -178,7 +186,7 @@
 #grid.index.url=http://training03.cagrid.org:6080/wsrf/services/DefaultIndexService
 #grid.index.url=http://cagrid-index-stage.nci.nih.gov:8080/wsrf/services/DefaultIndexService
 # Development Grid
-grid.index.url=http://cbiovdev5012.nci.nih.gov:8080/wsrf/services/DefaultIndexService
+grid.index.url=http://index.training.cagrid.org:8080/wsrf/services/DefaultIndexService
 grid.poc.tech.researchCenter.displayname=CBIIT
 grid.poc.tech.researchCenter.shortname=CBIIT
 grid.poc.tech.addr.country=USA
@@ -205,8 +213,16 @@
 grid.secure.key.location=${security.dist.relative.dir}/165.112.132.171-key.pem
 grid.external.secure.host=
 grid.external.secure.port=
+# This is used to download the targets for the grid, it is used by the *.grid.secure functionality.  May get
the grid.index.url from here in the future
+#grid.target=nci_prod-1.3
+#grid.target=nci_qa-1.3
+#grid.target=nci_stage-1.3
+#grid.target=training-1.3
+grid.target=training-1.3

+jboss.http-connector.remove=true

+
 ###################################
 #  PRE-POST VALIDATION PROPERTIES #
 ###################################
Index: project.properties
===================================================================
--- project.properties        (revision 7040)
+++ project.properties        (revision 7192)
@@ -14,7 +14,7 @@
 ignore.check.database=true
 require.build.wscore=true

-bda.version=0.10.4
+bda.version=0.10.9

 # Must correspond to versions specified w/in ../po/services/pom.xml
 po-services.version=3.0-SNAPSHOT
@@ -61,6 +61,12 @@
 ws-core.binaries.relative.dir=ws-core-4.0.3
 wscore.relative.dir=ws-core-4.0.3

+sync-gts.binaries.file=gaards-syncgts-1.3.0.1-bin.zip
+sync-gts.src.url=http://software.cagrid.org/gaards/1.3.0.1/${sync-gts.binaries.file}
+sync-gts.binaries.relative.dir=.
+#$SYNCGTS_LOCATION needs to be set in env to point to extracted location
+cagrid-target.src.url=https://ncisvn.nci.nih.gov/svn/cagrid/branches/caGrid-1_3_release/cagrid-1-0/caGrid
```

```
/repository/caGrid/target_grid
+
 findbugs.binaries.file=findbugs-1.3.4.zip
 findbugs.src.url=http://gforge.nci.nih.gov/svnroot/commonlibrary/trunk/other/os-independent/${findbugs.
binaries.file}
 findbugs.binaries.relative.dir=findbugs-1.3.4
@@ -78,6 +84,7 @@
 db.dist.relative.dir=db
 db-install.dist.relative.dir=db/db-install
 db-upgrade.dist.relative.dir=db/db-upgrade
+sync-gts.dist.relative.dir=sync-gts

 #*******************************
 # Databases build/install properties
Index: upgrade.properties
===================================================================
--- upgrade.properties        (revision 7040)
+++ upgrade.properties        (revision 7192)
@@ -51,11 +51,18 @@
 po-grid.jboss.server.jndi.port=21299
 po-grid.jboss.server.port=29280
 #used to allow the po-grid to make EJB3 calls via jndi
-pogrid.jndi.principal=ejbclient
-pogrid.jndi.credentials=pass
-pogrid.jboss.external.http.host=localhost
-pogrid.jboss.external.http.port=29280
+pogrid.jndi.principal=Gr1DU5er
+pogrid.jndi.credentials=Pa44Wurd
+pogrid.jboss.external.http.host=${pogrid.grid.external.secure.host}
+pogrid.jboss.external.http.port=${pogrid.grid.secure.port}

+pogrid.grid.secure.enable=true
+pogrid.grid.secure.port=29443
+pogrid.grid.secure.cert.location=${user.home}/.cagrid/certificates/${pogrid.grid.external.secure.host}-cert.pem
+pogrid.grid.secure.key.location=${user.home}/.cagrid/certificates/${pogrid.grid.external.secure.host}-key.pem
+pogrid.grid.external.secure.host=${env.HOSTNAME}
+pogrid.grid.external.secure.port=29443
+
 po-grid-legacy.jboss.server.name=pogridlegacy
 po-grid-legacy.jboss.server.jndi.port=21099
 po-grid-legacy.jboss.server.port=29080
@@ -76,7 +83,7 @@
 #grid.index.url=http://training03.cagrid.org:6080/wsrf/services/DefaultIndexService
 #grid.index.url=http://cagrid-index-stage.nci.nih.gov:8080/wsrf/services/DefaultIndexService
 # Development Grid
-grid.index.url=http://cbiovdev5012.nci.nih.gov:8080/wsrf/services/DefaultIndexService
+grid.index.url=http://index.training.cagrid.org:8080/wsrf/services/DefaultIndexService
 grid.poc.tech.researchCenter.displayname=CBIIT
 grid.poc.tech.researchCenter.shortname=CBIIT
 grid.poc.tech.addr.country=USA
@@ -103,7 +110,15 @@
 grid.secure.key.location=${security.dist.relative.dir}/165.112.132.171-key.pem
 grid.external.secure.host=
 grid.external.secure.port=
+# This is used to download the targets for the grid, it is used by the *.grid.secure functionality.  May get
the grid.index.url from here in the future
+#grid.target=nci_prod-1.3
+#grid.target=nci_qa-1.3
+#grid.target=nci_stage-1.3
+#grid.target=training-1.3
+grid.target=training-1.3

+jboss.http-connector.remove=true
+
 ###############
 ###  LDAP  ###
 ###############
Index: install.xml
===================================================================
--- install.xml        (revision 7040)
+++ install.xml        (revision 7192)
```

```
@@ -85,7 +85,13 @@
   <!-- Jboss configurtion related properties -->
   <property name="jboss.binding.template.location" value="${bda-utils.dir}/resource/${jboss.template.relative.
dir}/bindings.xml"/>
   <property name="jboss.service.template.location" value="${bda-utils.dir}/resource/${jboss.template.relative.
dir}/jboss-service.xml"/>
+  <!-- added for updated secure grid ssaksa 090826 -->
+   <property name="sync-gts.dir" location="${basedir}/${sync-gts.dist.relative.dir}"/>
+   <property name="sync-gts.build.dir" location="${sync-gts.dir}/syncgts"/>
+   <property name="cagrid-target.dir" location="${sync-gts.dir}/cagrid-target"/>
+   <property name="grid.dir.dest.jboss" value="wsrf.war" />

+
   <!-- *-ds.xml and WAR -->
   <property name="po-ear.dir.dist" value="${basedir}/${po-ear.dist.relative.dir}" />
   <property name="po-ear.ds.file" value="po-ds.xml" />
@@ -93,6 +99,7 @@
   <property name="po-ear.hibernate.file" value="hibernate.cfg.xml" />
   <property name="po-ear.ear.file" value="po.ear" />

+
   <!-- Default to false, properties can override -->
   <property name="grid.secure.enable" value="false"/>
   <property name="jboss.ssl.enable" value="false"/>
@@ -203,7 +210,11 @@
   <basename file="${grid.secure.cert.location}" property="grid.secure.cert.file"/>
   <dirname file="${grid.secure.cert.location}" property="grid.secure.dir"/>
   <basename file="${grid.secure.key.location}" property="grid.secure.key.file"/>
-
+
+  <basename file="${pogrid.grid.secure.cert.location}" property="pogrid.grid.secure.cert.file"/>
+  <dirname file="${pogrid.grid.secure.cert.location}" property="pogrid.grid.secure.dir"/>
+  <basename file="${pogrid.grid.secure.key.location}" property="pogrid.grid.secure.key.file"/>
+
   <!-- There is any issue with copying files with a filtersfile, any properties with a value of
     another property do not get expanded (xx=${yy} @xx@ will be replaced with ${yy} not the
     value).  I have defined a filter set below for these properties, I then two two copies
@@ -494,6 +505,8 @@
     <!-- Configures installed po-grid application -->
     <target name="install:po-grid:configure" description="Configure po-grid service based on properties"
unless="exclude.po-grid">
        <grid-appserver-configure
+         appserver.home="${jboss.home}"
+         appserver.server.name="${po-grid.jboss.server.name}"
          appserver.conf.dir="${jboss.home}/server/${po-grid.jboss.server.name}/conf"
          appserver.webapp.dir="${jboss.home}/server/${po-grid.jboss.server.name}/deploy"
          appserver.server-xml.file="${jboss.home}/server/${po-grid.jboss.server.name}/deploy/jbossweb-
tomcat55.sar/server.xml"
@@ -505,7 +518,13 @@
          search.port="8080"
          grid.application.name="${po-grid.introduce.skeleton.service.name}"
          grid.application.relative.dir="${po-grid.dir.target}"
-         grid.secure.enable="false"
+         grid.secure.dir="${pogrid.grid.secure.dir}"
+         grid.secure.enable="${pogrid.grid.secure.enable}"
+         grid.secure.port="${pogrid.grid.secure.port}"
+         grid.secure.key.file="${pogrid.grid.secure.key.file}"
+         grid.secure.cert.file="${pogrid.grid.secure.cert.file}"
+         grid.external.secure.host="${pogrid.grid.external.secure.host}"
+         grid.external.secure.port="${pogrid.grid.external.secure.port}"
          appserver.external.http.host="${pogrid.jboss.external.http.host}"
        />
     </target>
@@ -587,6 +606,7 @@
            grid.secure.key.file="${grid.secure.key.file}"
            grid.secure.cert.file="${grid.secure.cert.file}"
            jboss.java.opts="${jboss.java.opts}"
+           jboss.http-connector.remove="false"
            />
   </target>
```

```
@@ -597,7 +617,7 @@
                   filtering="true"
                   overwrite="true">
                   <filterset begintoken="@" endtoken="@">
-                      <filter token="jboss.server.port" value="@po-grid.jboss.server.port@"/>
+                      <filter token="jboss.server.port" value="@pogrid.grid.secure.port@"/>
                       <filter token="jboss.ejbinvoker.port" value="@po-grid.jboss.ejbinvoker.port@"/>
                       <filter token="jboss.server.rmi.port" value="@po-grid.jboss.server.rmi.port@"/>
                       <filter token="jboss.server.jndi.port" value="@po-grid.jboss.server.jndi.port@"/>
@@ -649,16 +669,17 @@
                   jboss.external.http.host="${pogrid.jboss.external.http.host}"
                   jboss.external.http.port="${pogrid.jboss.external.http.port}"
                   proxy.update.connector.port.http="8080"
-                  grid.external.secure.host="${grid.external.secure.host}"
-                  grid.external.secure.port="${grid.external.secure.port}"
+                  grid.external.secure.host="${pogrid.grid.external.secure.host}"
+                  grid.external.secure.port="${pogrid.grid.external.secure.port}"
                   jboss.server.hostname="${jboss.server.hostname}"
-                  jboss.grid.configure="false"
-                  grid.secure.dir="${grid.secure.dir}"
-                  grid.secure.enable="${grid.secure.enable}"
-                  grid.secure.port="${grid.secure.port}"
-                  grid.secure.key.file="${grid.secure.key.file}"
-                  grid.secure.cert.file="${grid.secure.cert.file}"
+                  jboss.grid.configure="true"
+                  grid.secure.dir="${pogrid.grid.secure.dir}"
+                  grid.secure.enable="${pogrid.grid.secure.enable}"
+                  grid.secure.port="${pogrid.grid.secure.port}"
+                  grid.secure.key.file="${pogrid.grid.secure.key.file}"
+                  grid.secure.cert.file="${pogrid.grid.secure.cert.file}"
                   jboss.java.opts="${jboss.java.opts}"
+                  jboss.http-connector.remove="true"
                   />
   </target>

@@ -731,6 +752,7 @@
                   grid.secure.key.file="${grid.secure.key.file}"
                   grid.secure.cert.file="${grid.secure.cert.file}"
                   jboss.java.opts="${jboss.java.opts}"
+                  jboss.http-connector.remove="false"
                   />
   </target>

Index: common/resources/grid/jboss-globus-libs-cagrid1_1.zip
===================================================================
Cannot display: file marked as a binary type.
svn:mime-type = application/octet-stream
Index: build.xml
===================================================================
--- build.xml        (revision 7040)
+++ build.xml        (revision 7192)
@@ -204,6 +204,14 @@
     </default>
   </switch>

+  <if>
+          <not>
+                  <isset property="env.ANT_OPTS"/>
+          </not>
+          <then>
+                  <fail message="To build this project you need to specify a increased java memory settings."/>
+          </then>
+  </if>
   <!-- Targets -->
   <target name="diagnostics" description="diagnostics">
     <echoproperties/>
@@ -447,9 +455,44 @@
     depends="
     init,
     dist:tools:retrieve:jboss,
-    dist:tools:retrieve:jboss-bindings
```

```
+    dist:tools:retrieve:jboss-bindings,
+    dist:tools:retrieve:sync-gts
     " />
+        <!--
+           See https://wiki.nci.nih.gov/display/BuildandDeploymentAutomation
/How+to+configure+a+Secure+Grid+Listener for more details
+        -->
+    <target name="dist:tools:retrieve:sync-gts"  description="Downloads caGrid SyncGTS service file from
binary repository and verifies checksum">
+            <if>
+                    <not>
+                            <available file="${download.dir}/${sync-gts.binaries.file}"/>
+                    </not>
+                    <then>
+                            <get src="${sync-gts.src.url}"     dest="${download.dir}/${sync-gts.binaries.
file}" />
+                    </then>
+            </if>
+            <property name="sync-gts.dist.dir" location="${dist.exploded.dir}/${sync-gts.dist.relative.dir}"/>
+            <unzip src="${download.dir}/${sync-gts.binaries.file}"     dest="${sync-gts.dist.dir}"/>
+            <property name="cagrid.target.co.dir" location="${target.dir}/cagrid-target"/>
+            <mkdir dir="${cagrid.target.co.dir}"/>
+            <if>
+                    <not>
+                            <available file="${cagrid-target.src.url}"/>
+                    </not>
+                    <then>
+                            <mkdir dir="${cagrid.target.co.dir}"/>
+                            <svn-co
+                                    svn.checkout.url="${cagrid-target.src.url}"
+                                    svn.checkout.dir="${cagrid.target.co.dir}"
+                                    delete="false"
+                                    />
+                    </then>
+            </if>
+            <copy todir="${sync-gts.dist.dir}/cagrid-target">
+                    <fileset dir="${cagrid.target.co.dir}"/>
+            </copy>
+    </target>

+
   <target name="dist:tools:retrieve:jboss-bindings"  description="Downloads JBOSS bindings filefrom binary
repository and verifies checksum">
     <get src="${jboss-bindings.src.url}"     dest="${dist.exploded.dir}/${jboss-bindings.file}" />
   </target>
@@ -513,7 +556,7 @@
   </target>

   <!-- Copies install time resources into distribution tree -->
-  <target name="dist:upgrader:prep">
+  <target name="dist:upgrader:prep" depends="dist:tools:retrieve:sync-gts">
     <!-- Copy po database scripts -->
     <copy todir="${dist.exploded.dir}/${db.dist.relative.dir}" overwrite="true">
       <fileset dir="${db.src.dir}">
```

It is important that the updated version of the `cog-jglobus.jar` is used. It is located within https://ncisvn.nci.nih.gov/svn/coppa/trunk/code/build-po/common/resources/grid/jboss-globus-libs-cagrid1_1.zip. Ideally these prepackaged WSRF & Globus archives should be within the IVY Repo or elsewhere within the BDA Repositories.

```
$ md5 cog-jglobus.jar
MD5 (cog-jglobus.jar) = ff78337a0af216fc946ad81fde1d0961
```

```
$ svn log  https://ncisvn.nci.nih.gov/svn/coppa/trunk/code/build-po/common/resources/grid/jboss-globus-libs-cagrid1_1.zip
------------------------------------------------------------------------
r7183 | smatyas | 2009-10-19 17:14:37 -0400 (Mon, 19 Oct 2009) | 1 line

PO-1292: adding security to po-grid; updates to bda-based builds; updates to poear security on remote endpoints; -pr slustbader
------------------------------------------------------------------------
r4198 | saksass | 2009-01-16 16:20:23 -0500 (Fri, 16 Jan 2009) | 1 line

Upgrade to bda-utils-0.10.0-beta with smatyas, plan to change to 0.9.1 when released shortly
------------------------------------------------------------------------
```

You should see the following changes within your grid instance's lib directory. Your local directory will be different than ours, 'po-grid'.

```
   A /trunk/code/po-grid/lib/antlr-2.7.6rc1.jar
   M /trunk/code/po-grid/lib/caGrid-core-1.3.jar
   A /trunk/code/po-grid/lib/caGrid-enforce-auth-extension-Service-1.3.jar
   M /trunk/code/po-grid/lib/caGrid-metadata-security-1.3.jar
   A /trunk/code/po-grid/lib/jaxen-1.1.jar
   A /trunk/code/po-grid/lib/jaxmejs-0.5.2.jar
   A /trunk/code/po-grid/lib/relaxngDatatype.jar
   A /trunk/code/po-grid/lib/xsom14.jar
```

Changes to Promotion Tiers (involves Systems Team)

- See Deployment Request for COPPA-PO 3.1 for more information to guide the discussion, https://ncisvn.nci.nih.gov/svn/coppa/trunk/documents/environment/3.1.0/COPPA-PO_DeploymentRequest.doc

Business Application Updates

None.

Grid Instance Updates

Warning: These updates only apply to the JBoss server instance that hosts new secure grid services

Overview of Updates

1. Request Host Certificates for each grid-related server instance that is to become secured.
2. Make updates to the various jboss-4.0.5.GA-jems-ejb3/server/<serverinstance>/deploy/jbossweb-tomcat55.sar/server.xml files for the JBoss server instances requiring a secure grid listener.
3. Make updates to server instance's bindings configuration (bindings.xml)
4. Ensure that OS user account has Globus available on the file system with a environment variable exported (export GLOBUS_LOCATION=<path>)

Request Host Certificates for each grid-related server instance that is to become secured.

Systems will need to request the host certificates for the various promotion tiers and place the generated pair of files (*-cert.pem and *-key.pem) in a location accessible to each of the various user accounts responsible for running each JBoss server instance by following the instructions here, http://cagrid.org/display/knowledgebase/Request+a+Host+Certificate.

When this is done a hostname will need to be specified which will be used by all server instances that resolve to this grid service hostname.

Make updates to the various jboss-4.0.5.GA-jems-ejb3/server/<serverinstance>/deploy/jbossweb-tomcat55.sar/server.xml files for the JBoss server instances requiring a secure grid listener.

Information on how to update jboss-4.0.5.GA-jems-ejb3/server/<serverinstance>/deploy/jbossweb-tomcat55.sar/server.xml files

The Grid uses it's own Trust Fabric and does not require certificates from an external Certificate Authority (CA) vendor, it includes it's own local CA and knows how to trust these certificates. This is not a standard SSL configuration.

Basically, in this step you'll be adding a HTTPS <Connector> and removing any existing HTTP & HTTPS <Connector>(s) for the <Service> definition within the bundled Tomcat servlet container inside JBoss.

In the original file you'll notice the proxyPort is set to the HTTPPort defined for the instance's specific binding configuration (see your bindings.xml) - 29080 in the example below

Before: jboss-4.0.5.GA-jems-ejb3/server/<serverinstance>/deploy/jbossweb-tomcat55.sar/server.xml

```
<Server>
   <Service>
   ...
     <Connector acceptCount="100"
          address="$\{jboss.bind.address\}"
          connectionTimeout="20000"
          disableUploadTimeout="true"
          emptySessionPath="true"
          enableLookups="false"
          maxHttpHeaderSize="8192"
          maxThreads="250"
          port="8080"
          proxyName="localhost"
          proxyPort="29080"
          redirectPort="8443"
          strategy="ms"/>
  ...
    <Engine>
      <Host>
  ...
      </Host>
    </Engine>
  </Service>
</Server>
```

Now, here is where things may get somewhat tricky (or not). To add the new Connector you'll need to remove the existing and add (define) a new <Connector>. Below is an example:

After: jboss-4.0.5.GA-jems-ejb3/server/<serverinstance>/deploy/jbossweb-tomcat55.sar/server.xml

```
<Server>
   <Service>
   ...
     <Connector acceptCount="10"
           autoFlush="true"
           cert="/Users/smatyas/apps/po/jboss-4.0.5.GA-jems-ejb3/server/<serverinstance>/conf
/<serverinstancehostname>-cert.pem"
           className="org.globus.tomcat.coyote.net.HTTPSConnector"
           debug="0"
           disableUploadTimeout="true"
           enableLookups="true"
           key="/Users/smatyas/apps/po/jboss-4.0.5.GA-jems-ejb3/server/<serverinstance>/conf
/<serverinstancehostname>-key.pem"
           maxSpareThreads="75"
           maxThreads="150"
           minSpareThreads="25"
           port="<DesiredPortForHTTPS>"
           protocolHandlerClassName="org.apache.coyote.http11.Http11Protocol"
           proxyName="<serverinstancehostname>"
           proxyPort="<DesiredPortForHTTPS>"
           scheme="https"
           socketFactory="org.globus.tomcat.catalina.net.BaseHTTPSServerSocketFactory"/>
  ...
    <Engine>
      <Host>
  ...
      </Host>
      <Valve className="org.globus.tomcat.coyote.valves.HTTPSValve55"/>
    </Engine>
  </Service>
</Server>
```

In the above example, you'll notice the absolute path to Host Cert files for the cert and key attributes. Again, these files can be anywhere on the filesystem so long as they are both accessible to the user account tied to the particular jboss server instance (jboss-4.0.5.GA-jems-ejb3/server/<serverinstance>/). Next, you'll need to make sure you choose a <DesiredPortForHTTPS> for both the port and proxyPort attributes and that they are the same.

Make updates to server instance's bindings configuration (bindings.xml)

Lastly, some changes will need to be made to the server instance bindings configuration for our instance's configuration. In short, since we've removed the existing HTTP-based <Connector> and replaced it with a HTTPS-based <Connector> we'll need to update the references to the previously defined HTTP-based port within the bindings.xml. Attached is an example bindings.xml that we've generated. You'll notice that we use 29443 throughout for our HTTPS port.

It may be easiest, though somewhat confusing, to simply repurpose the existing HTTP port to become the HTTPS port. We choose not to do that however, that appears to be a viable option too. If you do this the AJP port will change make sure to adjust the ports to exactly how you want them.

Ensure that OS user account has Globus available on the file system with a environment variable exported (export GLOBUS_LOCATION=<path>)

The binary can be found here, http://gforge.nci.nih.gov/svnroot/commonlibrary/trunk/techstack-2006/os-independent/ws-core-enum-4.0.3.zip

We recommend that the WS-CORE-4.0.3 be unpacked into a shared directory, say /usr/local/ws-core-4.0.3. Then, update the user's bash profile (~/.bash_profile) to define and export GLOBUS_LOCATION.

After: ~/.bash_profile

```
ANT_HOME=/local/home/jboss45e/apache-ant-1.7.0
JAVA_HOME=/usr/jdk1.5.0_10
GLOBUS_LOCATION=/usr/local/ws-core-4.0.3

export ANT_HOME JAVA_HOME GLOBUS_LOCATION
export PATH=$ANT_HOME/bin:$JAVA_HOME/bin:$PATH
```

Ensure Globus libs are in place

If all the changes have been made but there is an issue with the globus class path. The libs are not available for the jboss server.
The following libs must be available.
Move the following files into the base /usr/local/jboss405/server/<app name>/lib

~~rw-r-r~~ 1 jboss45 jboss45 23686 Dec 10 15:39 cog-tomcat.jar
~~rw-r-r~~ 1 jboss45 jboss45 659777 Dec 10 15:39 cog-jglobus.jar