
GT 4.2.1 Migrating Guide for Java WS A&A

Table of Contents

1. Migrating Java WS Authorization Framework from GT 4.0	1
2. Migrating Java WS Authorization Framework from GT3	3
3. Migrating host credentials from GT3 and GT2	3
Glossary	3

<titleabbrev>Migrating Guide</titleabbrev>

The following provides available information about migrating from previous versions of the Globus Toolkit.

1. Migrating Java WS Authorization Framework from GT 4.0

Java WS Authorization framework has been reworked with WS independent authorization engine and separate authorization module. The following sections describe migrating from GT 4.0.x and intermediate GT 4.1.x development releases.

1.1. Interface and code changes

The Java WS Authorization Framework now uses the generic GT Java Authorization Framework, that eliminates dependency on web services components. The following changes will be needed from GT 4.0.x:

1. Package name: PDP/PIP interfaces, attribute processing classes and configuration classes are now used from the generic authorization engine. Hence the authorization interfaces have changed from `org.globus.wsrp.security.authorization` to `org.globus.security.authorization`. The following classes have changed:
 - Attribute
 - AttributeCollection
 - AttributeException
 - AttributeIdentifier
 - AuthorizationConfig
 - AuthorizationDeniedException
 - AuthorizationEngineSpi
 - AuthorizationException
 - BootstrapPIP
 - ChainConfig

- CloseException
 - Decision
 - EntityAttributes
 - IdentityAttributeCollection
 - InitializeException
 - Interceptor
 - InterceptorConfig
 - InterceptorException
 - PDP
 - PIP
2. RequestAttributes class: This class has been renamed as RequestEntities with no functionality change
 3. PIPResponse class: This class has been renamed as NonRequestEntities. Functionality from the older class has been preserved, with additional methods to merge attributes to this structure has been added.
 4. PIP Interface change: The new PIP interface is as follows:

```
public
    NonRequestEntities collectAttributes(RequestEntities requestAttr) throws AttributionException;
```

MessageContext has been removed from the interface. Refer to item (7) on information on retrieving message context.

This interface does not extend from Interceptor interface. But PIPInterceptor interface is equivalent to the previous version of the PIP interface, with collect attributes method and interceptor interface methods.

5. PDP Interface change: The new PDP interface is as follows:

```
public Decision
    canAccess(RequestEntities requestEntities, NonRequestEntities nonReqEntities) throws AuthorizationException;
    public Decision canAdminister(RequestEntities requestEntities, NonRequestEntities nonReqEntities) throws AuthorizationException;
```

NonRequestAttributes class encompasses the three List objects for non-request subject, resource and action.

The RequestAttributes class is replaced by RequestEntities class as is. MessageContext has been removed from the interface. Refer to item (7) on information on retrieving message context.

This interface does not extend from Interceptor interface. But PDPInterceptor interface is equivalent to the previous version of the PDP interface, with collect attributes method and interceptor interface methods.

6. Providers package: The providers that were a part of the authorization package are now a part of the generic interface. So the FirstApplicable and PermitOverride combining algorithm interface, in addition to the AbstractEngine class are now a part of the new package, org.globus.security.authorization.providers.

7. Message Context: ContainerPIP, the default PIP that is used by the GT framework to initialize request context, adds the message context associated with the request as an environment attribute with null issuer. To extract the message context, the following code snippet can be used:

```
RequestEntities reqEntities; org.apache.axis.MessageContext msgCtx =  
    AttributeUtil.getMessageContext(reqEntities.getEnvironment(), null);
```

1.2. Authorization Module Changes

The Java WS server side authorization code has been moved to a separate module called `authorization`. A migration guide, that outlines the changes needed for services that build on Java WS Core, is provided [here](#).¹

2. Migrating Java WS Authorization Framework from GT3

While the GT4 version of this component has similar features to the GT3 version, some of the configuration methodology has changed and some features have been enhanced. Refer to [Section 5, "Authorization"](#) for changes in configuration.

3. Migrating host credentials from GT3 and GT2

GT2 and GT3 services were set up to run with root owned *host credentials*. In GT4 most, but not all, services will run as the **globus** user. To allow the **globus** user to start services using host credentials, the **globus** user needs to be able to access them. This requirement can be satisfied by making a copy of the root-owned host credentials, i.e. the *host certificate* and *private key*, owned by the **globus** user. In GT4 this copy is assumed to be `/etc/grid-security/container{cert,key}.pem`.

Glossary

H

host certificate An EEC belonging to a host. When using GSI this certificate is typically stored in `/etc/grid-security/hostcert.pem`. For more information on possible host certificate locations see the [GSI C Developer's Guide](#).

host credentials The combination of a host certificate and its corresponding private key.

P

private key The private part of a key pair. Depending on the type of certificate the key corresponds to it may typically be found in `$HOME/.globus/userkey.pem` (for user certificates), `/etc/grid-security/hostkey.pem` (for host certificates) or `/etc/grid-security/<service>/<service>key.pem` (for service certificates).

For more information on possible private key locations see [this](#).

¹ http://dev.globus.org/wiki/Java_WS_Core/Independent_Java_Authz_Module