

# GT4 C WS A&A Admin Guide

DRAFT

# GT4 C WS A&A Admin Guide

## Introduction

This guide contains advanced configuration information for system administrators working with C WS A&A. It provides references to information on procedures typically performed by system administrators, including installing, configuring, deploying, and testing the installation.

### Important

C WS A&A is built, installed and deployed as part of C WS Core - which is part of a default GT installation. See [Installing GT 4.2.0](#) for installation details.

The main administration issues for this component deal with configuring credential-related settings. There are multiple mechanisms for doing this:

- Security Descriptors (This is the *preferred* mechanism)
  - Container Security Descriptor
  - Service Security Descriptor
- CoG properties
- Environment variables
- Relying on default behavior. The only default behaviors available concern the proxy file and trusted certificates locations.

More information on these mechanisms can be found in the [public interface guide](#).

## Table of Contents

1. Building and Installing .....	1
2. Configuring .....	2
1. Configuration overview .....	2
2. Syntax of the interface .....	2
3. Deploying .....	4
4. Testing .....	5
5. Security Considerations .....	6
1. Security considerations for C WS A&A .....	6
6. Debugging .....	7
1. Logging .....	7
7. Troubleshooting .....	8
1. Error Messages For C WS A&A .....	9
2. Credential Troubleshooting .....	13
Glossary .....	16

## List of Tables

2.1. Configuring server side authentication and message/transport security .....	3
7.1. C WS A&A Errors .....	10
7.2. Credential Errors .....	14

DRAFT

---

# Chapter 1. Building and Installing

The GT4 C WS A&A component is currently installed as part of the GT4 C WS Core component. More information on installing this component can be found in the "Building and Installing" section of the [Java WS Core Admin Guide](#).

DRAFT

# Chapter 2. Configuring

## 1. Configuration overview

Configuration of service-side security settings can be achieved by using container or service security descriptor. Some of the security configuration, like the credential to use and trusted certificates location, can also be configured using CoG properties or rely on default location. **The preferred way is to provide these settings in a security descriptor.**

The next section provides details on the relevant properties. An overview of the syntax of security descriptors can be found in [Java WS A&A Security Descriptor Framework](#). Available CoG security properties can be found in [Chapter 2, Configuring](#)

## 2. Syntax of the interface

The following properties are relevant to authentication and message/transport security:

**Table 2.1. Configuring server side authentication and message/transport security**

Number	Task	Descriptor Configuration	Alternate Configuration
1	Credentials	<u>Container or service descriptor configuration</u> <sup>1</sup>	<ul style="list-style-type: none"> <li>• X509_USER_CERT or <u>CoG Configuration</u><sup>2</sup>: User certificate configuration</li> <li>• X509_USER_KEY or <u>CoG Configuration</u><sup>3</sup>: User key configuration</li> <li>• X509_USER_PROXY or <u>CoG Configuration</u><sup>4</sup>: User proxy configuration</li> </ul> <p>If no explicit configuration is found, the default proxy is read from /tmp/x509_up_&lt;uid&gt;.</p>
2	Trusted Certificates	<u>Container security descriptor configuration</u> <sup>5</sup>	<u>CoG Configuration</u> <sup>6</sup>
3	Limited proxy policy configuration	<u>Container or service descriptor configuration</u> <sup>7</sup>	None.
4	Replay Attack Window	<u>Container or service descriptor configuration</u> <sup>8</sup>	None.
5	Replay Attack Filter	<u>Container or service descriptor configuration</u> <sup>9</sup>	None.
6	Replay timer interval	<u>Container descriptor configuration</u> <sup>10</sup>	None.
7	Context timer interval	<u>Container descriptor configuration</u> <sup>11</sup>	None.

<sup>1</sup> <http://www.globus.org/toolkit/docs/4.2/4.2.0/security/wsaajava/wsaajava-secdesc.html#wsaajava-secdesc-configCred>

<sup>2</sup> <http://www.globus.org/toolkit/docs/4.2/4.2.0/common/javacog/admin/javacog-admin-configuring.html#javacog-admin-configuring-user-certificate>

<sup>3</sup> <http://www.globus.org/toolkit/docs/4.2/4.2.0/common/javacog/admin/javacog-admin-configuring.html#javacog-admin-configuring-user-key>

<sup>4</sup> <http://www.globus.org/toolkit/docs/4.2/4.2.0/common/javacog/admin/javacog-admin-configuring.html#javacog-admin-configuring-user-proxy>

<sup>6</sup> <http://www.globus.org/toolkit/docs/4.2/4.2.0/common/javacog/admin/javacog-admin-configuring.html#javacog-admin-configuring-trusted-certs>

<sup>5</sup> <http://www.globus.org/toolkit/docs/4.2/4.2.0/security/wsaajava/wsaajava-secdesc.html#wsaajava-secdesc-container-trusted>

<sup>7</sup> <http://www.globus.org/toolkit/docs/4.2/4.2.0/security/wsaajava/wsaajava-secdesc.html#wsaajava-secdesc-rejectLimProxy>

<sup>8</sup> <http://www.globus.org/toolkit/docs/4.2/4.2.0/security/wsaajava/wsaajava-secdesc.html#wsaajava-secdesc-replayAttack>

<sup>9</sup> <http://www.globus.org/toolkit/docs/4.2/4.2.0/security/wsaajava/wsaajava-secdesc.html#wsaajava-secdesc-replayAttack>

<sup>10</sup> <http://www.globus.org/toolkit/docs/4.2/4.2.0/security/wsaajava/wsaajava-secdesc.html#wsaajava-secdesc-container-replay>

<sup>11</sup> <http://www.globus.org/toolkit/docs/4.2/4.2.0/security/wsaajava/wsaajava-secdesc.html#wsaajava-secdesc-container-context>

# Chapter 3. Deploying

The GT4 C WS A&A component is currently deployed as part of the GT4 C WS Core component.

DRAFT

# Chapter 4. Testing

FIXME - information for testing c ws security.

DRAFT

# Chapter 5. Security Considerations

## 1. Security considerations for C WS A&A

### 1.1. File permissions

The Java security code currently does not enforce secure permissions and, implicitly, file ownership requirements on any of the security related files, e.g. configuration and credential files. It is thus important that administrators ensure that the relevant files have correct permissions and ownership. Permissions should generally be as restrictive as possible, i.e. *private keys* should be readable only by the file owner and other files should be writable by owner only, and the files should generally be owned by the globus user (the requirements that the C code enforces are documented in [Configuring GSI](#)).

Also refer to [Section 5, “Known Problems”](#) for details on any other open issues.

# Chapter 6. Debugging

Because this component is built on C WS Core, it uses the same sys admin logging, described below:

## 1. Logging

As of 4.2.0, the Globus Toolkit provides system administration logs that are [CEDPs best practices](http://cedps.net/index.php/LoggingBestPractices)<sup>1</sup> compliant.

To enable CEDPS logging, pass the `-log PATH` parameter to the `globus-wsc-container` program.

For more details on the CEDPS Logging format, including descriptions of reserved name-value pairs, see <http://cedps.net/index.php/LoggingBestPractices>:

### 1.1. Sample log file

The [sample log file](http://www.globus.org/toolkit/docs/4.2/4.2.0/common/cwscore/sample-container-log.txt)<sup>2</sup> contains many log entries for various scenarios in the C WS container.

---

<sup>1</sup> <http://cedps.net/index.php/LoggingBestPractices>

<sup>2</sup> <http://www.globus.org/toolkit/docs/4.2/4.2.0/common/cwscore/sample-container-log.txt>

# Chapter 7. Troubleshooting

For a list of common errors in GT, see [Error Codes](#).

DRAFT

# 1. Error Messages For C WS A&A

DRAFT

**Table 7.1. C WS A&A Errors**

Error Code	Definition
<pre>ERROR: Couldn't read user key: Bad passphrase key file location: /Users/bester/.globus/userkey.pem  globus_credential: Error reading user credential: Can't read credential's private key from PEM OpenSSL Error: pem_lib.c:423: in library: PEM routines, function PEM_do_header: bad decrypt OpenSSL Error: evp_enc.c:509: in library: digital envelope routines, function EVP_DecryptFinal: bad decrypt  Use -debug for further information.</pre>	Unable to decrypt private key
<pre>globus_gsi_gssapi: Error with gss credential handle globus_credential: Valid credentials could not be found in any of the possible locations specified by the credential search order. Valid credentials could not be found in any of the possible locations specified by the credential search order. Attempt 1 globus_credential: Error reading host credential globus_sysconfig: Error with certificate filename globus_sysconfig: Error with certificate filename globus_sysconfig: File is not owned by current user: /etc/grid-security/hostcert.pem is not owned by current user Attempt 2 globus_credential: Error reading proxy credential globus_sysconfig: Could not find a valid proxy certificate file location globus_sysconfig: Error with key filename globus_sysconfig: File does not exist: /tmp/x509up_u501 is not a valid file Attempt 3 globus_credential: Error reading user credential globus_credential: Key is password protected: GSI does not currently support password protected private keys. OpenSSL Error: pem_lib.c:401: in library: PEM routines, function PEM_do_header: bad password read</pre>	No user proxy could be found
<pre>globus_gsi_gssapi: Error with GSI credential globus_gsi_gssapi: Error with gss credential handle globus_credential: Error with credential: The proxy credential: /tmp/x509up_u1499     with subject: /DC=org/DC=example/DC=grid/OU=People/CN=Joe User/CN=1235439010     expired 44 minutes ago.</pre>	Proxy has expired.

Error Code	Definition
<p>globus_xio: The GSI XIO driver failed to establish a secure connection. The failure occurred during a handshake read.  globus_xio: An end of file occurred</p>	<p>Communication disrupted during SSL handshake</p>
<p>globus_gsi_gssapi: Unable to verify remote side's credentials  globus_gsi_gssapi: Unable to verify remote side's credentials: Couldn't verify the remote certificate  OpenSSL Error: s3_pkt.c:1052: in library: SSL routines, function SSL3_READ_BYTES: sslv3 alert bad certificate SSL alert number 42</p>	<p>Unable to verify remote certificate. Often a clock-synchronization problem where the service clock is behind that of the client.</p>
<p>OpenSSL Error: s3_clnt.c:894: in library: SSL routines, function SSL3_GET_SERVER_CERTIFICATE: certificate verify failed  globus_gsi_callback_module: Could not verify credential  globus_gsi_callback_module: The certificate is not yet valid: Cert with subject: /DC=org/DC=example/DC=grid/OU=People/CN=Joe User/CN=464555355 is not yet valid- check clock skew between hosts.</p>	<p>Unable to verify remote certificate. Often a clock-synchronization problem where the client clock is behind that of the service.</p>

Error Code	Definition
<pre> globus_gsi_callback_module: Error with signing policy globus_sysconfig: Error getting signing policy file globus_sysconfig: File does not exist: /etc/grid-security/certificates/2b0e42b2.signing_policy is not a valid file </pre>	<p>The service's certificate is not trusted by the client</p>
<pre> globus_gsi_callback_module: Could not verify credential globus_gsi_callback_module: Error with signing policy globus_gsi_callback_module: Error in OLD GAA code: CA policy violation: &lt;no reason given&gt; </pre>	<p>Service certificate is not trusted because the CA signing policy does not trust the CA to sign the subject name of the certificate.</p>
<pre> Error: globus_soap_message_module: SOAP Fault Fault code: Client Fault string: globus_handler_ws_secure_message: Server Request handling failed globus_handler_ws_secure_message: Failed to verify the message: Unable to get Security header element from message attributes. </pre>	<p>The client sent a request to a service which message security without properly invoking the security handlers</p>

Error Code	Definition
<pre>Error: globus_soap_message_module: SOAP Fault Fault code: Client Fault string: globus_soap_message_module: Loaded message handlers do not understand required header element: {http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd} Security</pre>	<p>The client sent a request protected with message-level security but the server did not understand the required security headers</p>

## 2. Credential Troubleshooting

### 2.1. Credential Errors

The following are some common problems that may cause clients or servers to report that credentials are invalid:

For a list of common errors in GT, see [Error Codes](#).

**Table 7.2. Credential Errors**

<b>Error Code</b>	<b>Definition</b>	<b>Possible Solutions</b>
Your proxy credential may have expired	Your proxy credential may have expired.	Use <code>grid-proxy-info</code> to check whether the proxy credential has actually expired. If it has, generate a new proxy with <code>grid-proxy-init</code> .
The system clock on either the local or remote system is wrong.	This may cause the server or client to conclude that a credential has expired.	Check the system clocks on the local and remote system.
Your end-user certificate may have expired	Your end-user certificate may have expired	Use <code>grid-cert-info</code> to check your certificate's expiration date. If it has expired, follow your CA's procedures to get a new one.
The permissions may be wrong on your proxy file	If the permissions on your proxy file are too lax (for example, if others can read your proxy file), Globus Toolkit clients will not use that file to authenticate.	You can "fix" this problem by changing the permissions on the file or by destroying it (with <code>grid-proxy-destroy</code> ) and creating a new one (with <code>grid-proxy-init</code> ).  <b>Important:</b> However, it is still possible that someone else has made a copy of that file during the time that the permissions were wrong. In that case, they will be able to impersonate you until the proxy file expires or your permissions or end-user certificate are revoked, whichever happens first.
The permissions may be wrong on your private key file	If the permissions on your end user certificate private key file are too lax (for example, if others can read the file), <code>grid-proxy-init</code> will refuse to create a proxy certificate.	You can "fix" this by changing the permissions on the private key file.  <b>Important:</b> However, you will still have a much more serious problem: it is possible that someone has made a copy of your private key file. Although this file is encrypted, it is possible that someone will be able to decrypt the private key, at which point they will be able to impersonate you as long as your end user certificate is valid. You should contact your CA to have your end-user certificate revoked and get a new one.
The remote system may not trust your CA	The remote system may not trust your CA	Verify that the remote system is configured to trust the CA that issued your end-entity certificate. See <a href="#">Installing GT 4.2.0</a> for details.
You may not trust the remote system's CA	You may not trust the remote system's CA	Verify that your system is configured to trust the remote CA (or that your environment is set up to trust the remote CA). See <a href="#">Installing GT 4.2.0</a> for details.
There may be something wrong with the remote service's credentials	There may be something wrong with the remote service's credentials	It is sometimes difficult to distinguish between errors reported by the remote service regarding your credentials and errors reported by the client interface regarding the remote service's credentials. If you cannot find anything wrong with your credentials, check for the same conditions on the remote system (or ask a remote administrator to do so).

## 2.2. Some tools to validate certificate setup

### 2.2.1. Check that the user certificate is valid

```
openssl verify -CApath /etc/grid-security/certificates
-purpose sslclient ~/.globus/usercert.pem
```

### 2.2.2. Connect to the server using s\_client

```
openssl s_client -ssl3 -cert ~/.globus/usercert.pem -key
~/.globus/userkey.pem -CApath /etc/grid-security/certificates
-connect <host:port>
```

Here `<host:port>` denotes the server and port you connect to.

If it prints an error and puts you back at the command prompt, then it typically means that the *server* has closed the connection, i.e. that the server was not happy with the client's certificate and verification. Check the SSL log on the server.

If the command "hangs" then it has actually opened a telnet style (but secure) socket, and you can "talk" to the server.

You should be able to scroll up and see the subject names of the server's verification chain:

```
depth=2 /DC=net/DC=ES/O=ESnet/OU=Certificate Authorities/CN=ESnet Root CA 1
verify return:1
depth=1 /DC=org/DC=DOEGrids/OU=Certificate Authorities/CN=DOEGrids CA 1
verify return:1
depth=0 /DC=org/DC=doegrids/OU=Services/CN=wiggum.mcs.anl.gov
verify return:1
```

In this case, there were no errors. Errors would give you an extra line next to the subject name of the certificate that caused the error.

### 2.2.3. Check that the server certificate is valid

Requires root login on server:

```
openssl verify -CApath /etc/grid-security/certificates -purpose sslserver
/etc/grid-security/hostcert.pem
```

# Glossary

*some terms not in the docs but wanted in glossary: [scheduler](#)*

## 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](#).

## S

scheduler

Term used to describe a job scheduler mechanism to which GRAM interfaces. It is a networked system for submitting, controlling, and monitoring the workload of batch jobs in one or more computers. The jobs or tasks are scheduled for execution at a time chosen by the subsystem according to an available policy and availability of resources. Popular job schedulers include Portable Batch System (PBS), Platform LSF, and IBM LoadLeveler.