

GT 4.2.0 DRS: System Administrator's Guide

DRAFT

GT 4.2.0 DRS: System Administrator's Guide

Introduction

This guide contains advanced configuration information for system administrators working with the Data Replication Service (DRS). It provides references to information on procedures typically performed by system administrators, including installing, configuring, deploying, and testing the installation.

Important

This information is in addition to the basic Globus Toolkit prerequisite, overview, installation, security configuration instructions in the [Installing GT 4.2.0](#). Read through this guide before continuing!

DRAFT

Table of Contents

1. Building and installing	1
1. Installing DRS using Globus Toolkit Binary Bundle	1
2. Installing DRS using Globus Toolkit Source Bundle	1
3. Installing DRS using Globus CVS	1
2. Configuring	3
1. Configuration overview	3
2. Syntax of the interface	3
3. Testing	4
4. Security considerations	5
1. Data Replication Service (DRS) Security Considerations	5
5. Debugging	6
1. Verbose log output	6
2. Logging in Java WS Core	6
6. Troubleshooting	8
1. Errors	8
Index	9

List of Tables

6.1. Data Replication Service (DRS) Errors 8

DRAFT

Chapter 1. Building and installing

DRS is distributed with the Globus Toolkit 4.2.0 and is available in both the binary bundles and the source bundle. For most purposes, the binary bundle provides the simplest means of installing the DRS and its requirements. There are three typical ways to access and install the DRS:

- [Section 1, “Installing DRS using Globus Toolkit Binary Bundle”](#)
- [Section 2, “Installing DRS using Globus Toolkit Source Bundle”](#)
- [Section 3, “Installing DRS using Globus CVS”](#)

Important

Before proceeding with installation of DRS, please familiarize yourself with the installation procedures for RLS (see [Replica Location Service \(RLS\)](#)) and RFT (see [Reliable File Transfer \(RFT\) Service](#)).

1. Installing DRS using Globus Toolkit Binary Bundle

Follow instructions provided by [Installing GT 4.2.0](#) and [Installing GT](#) in order to install the DRS from one of the available binary bundles of the Globus Toolkit.

Once you have unpackaged the binary bundle, the following commands may be used to install DRS:

```
% ./configure --prefix=$GLOBUS_LOCATION --enable-rls --enable-drs
% make
% make install
```

2. Installing DRS using Globus Toolkit Source Bundle

In order to install DRS, you will need to ensure that RLS is also installed as part of the build by reviewing its installation procedures. The following commands will install the DRS (optionally you may use the "drs" make target to build only DRS and its requirements:

```
% ./configure --prefix=/path/to/install --enable-rls --enable-drs
% make [drs]
% make install
```

3. Installing DRS using Globus CVS

Additionally, you may access the DRS from the Globus Toolkit CVS repository. You may deploy the 4.0.x DRS on top of an existing GT 4.0.x installation using the `globus_4_0_branch` tag. You may deploy the 4.1.x DRS on top of an existing GT 4.2 installation using the `globus_4_2_branch` tag.

The following example instructions will only work given the presence of an existing GT installation with *all DRS dependencies*. The best way to establish such an environment is to first follow the basic installation instructions and subsequently use the following instructions to obtain the latest updates from the CVS branch. Note the use of the `globus_X_Y_branch` tag depends on whether you are interested in updates from the maintenance branch or the development trunk.

```
% setenv GLOBUS_LOCATION /path/to/existing/gtXY/install/  
% setenv CVSROOT :pserver:anonymous@cvs.globus.org:/home/globdev/CVS/globus-packages  
% cvs co [-r globus_X_Y_branch] ws-replica/replicator/  
% cd ws-replica/replicator/  
% ant deploy
```

To learn more about general instructions regarding GT CVS access, see [Remote CVS Access](#)¹.

¹ <http://www.globus.org/toolkit/docs/development/remote-cvs.html>

Chapter 2. Configuring

This information is in addition to the basic configuration instructions in the [Installing GT 4.2.0](#). Aside from the basic configuration of GT 4.2.0, please review the following instructions:

1. Configuration overview

The DRS requires certain JNDI settings to be properly configured. The installed JNDI configuration file may be found at `$GLOBUS_LOCATION/etc/globus_wsrp_replicator/jndi-config.xml`. To view the default configuration file (shipped with the GT 4.2.0 release) from the Globus CVS repository [click here](#)¹.

2. Syntax of the interface

The settings are structured as name-value pairs. For example:

```
<parameter>
  <name>defaultIndexUrl</name>
  <value>rls://127.0.0.1:39281</value>
</parameter>
```

The following settings must be properly configured:

- `proxyfileDir`: the directory that you would like the DRS to temporarily store user proxies. No setting is necessary. This value may be empty.
- `requestfileDir`: the directory that you would like the DRS to temporarily store request files. No setting is necessary. This value may be empty.
- `defaultIndexUrl`: the connection URL for your installation of RLS running as a RLI service.
- `defaultRegistrationUrl`: the connection URL for your installation of RLS running as a LRC service.
- `defaultReliableTransferUrl`: the connection URL for your installation of the RFT ReliableFileTransferFactoryService.
- `proxyfileChangePermsCmd`: the platform-dependent command to change file permissions to user-only read-write permissions.
- The rest of the parameter/value pairs may retain the given default values.

¹ http://viewcvs.globus.org/viewcvs.cgi/ws-replica/replicator/service/java/source/deploy-jndi-config.xml?tab=globus_4_2_branch&content-type=text/vnd.viewcvs-markup

Chapter 3. Testing

This service does not provide a set of tests yet.

DRAFT

Chapter 4. Security considerations

1. Data Replication Service (DRS) Security Considerations

1.1. Service configuration files

The service configuration files such as the JNDI configuration file, `jndi-config.xml`, and the Web service deployment descriptor, `server-config.wsdd`, located in the `$GLOBUS_LOCATION/etc/globus_wsrf_replicator` directory, contain sensitive information such as database username and password. It is important to ensure that these files are readable only by the system administrator that is responsible for the container. During deployment, the permissions on these files are adjusted automatically, however, you should verify the permissions to ensure that they have been correctly set for your specific platform.

1.2. Delegated proxy credential files

Creating a Replicator requires that the user supply a delegated credential to the DRS during the initial creation request. The service retrieves the delegated credential from the Delegation Service and stores it on the file system. As part of the DRS configuration (see installation and configuration instructions), the user selects a directory to use for storage of delegated credentials. The default setting is for the DRS to store the file in the system's designated temporary directory (e.g., `/tmp` on many platforms). The service sets the permissions on the temporary file such that it can only be accessed by the user account used to run the container.

Chapter 5. Debugging

1. Verbose log output

Generating verbose log output is a critical aid in troubleshooting of the DRS and is useful when communicating problems to Globus support lists. To increase logging detail, add the following line to the `$GLOBUS_LOCATION/container-log4j.properties` file.

```
...
log4j.category.org.globus.replica=DEBUG
...
```

2. Logging in Java WS Core

The following information applies to Java WS Core and all services built on Java WS Core.

Java WS Core server side has two types of loggers. One logger is used for development logging and by default writes to standard out. The other logger includes system administration information and is [CEDPs best practices](#)¹ compliant.

On client side, only developer logging is available and is configured using `log4j.properties`.

2.1. Development Logging in Java WS Core

The following information applies to Java WS Core and those services built on it.

Logging in the Java WS Core is based on the [Jakarta Commons Logging](#)² API. Commons Logging provides a consistent interface for instrumenting source code while at the same time allowing the user to plug-in a different logging implementation. Currently we use [Log4j](#)³ as a logging implementation. Log4j uses a separate configuration file to configure itself. Please see Log4j documentation for details on the [configuration file format](#)⁴.

2.1.1. Configuring server side developer logs

Server side logging can be configured in `$GLOBUS_LOCATION/container-log4j.properties`, when the container is stand alone container. For tomcat level logging, refer to [Logging for Tomcat](#)⁵. The logger `log4j.appender.A1` is used for developer logging and by default writes output to the system output. By default it is set for all warnings in the Globus Toolkit package to be displayed.

Additional logging can be enabled for a package by adding a new line to the configuration file. Example:

```
#for debug level logging from org.globus.package.FooClass
log4j.category.org.globus.package.name.FooClass=DEBUG
#for warnings from org.some.warn.package
```

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

² <http://jakarta.apache.org/commons/logging/>

³ <http://logging.apache.org/log4j/>

⁴ [http://logging.apache.org/log4j/docs/api/org/apache/log4j/PropertyConfigurator.html#doConfigure\(java.lang.String,org.apache.log4j.spi.LoggerRepository\)](http://logging.apache.org/log4j/docs/api/org/apache/log4j/PropertyConfigurator.html#doConfigure(java.lang.String,org.apache.log4j.spi.LoggerRepository))

⁵ <http://tomcat.apache.org/tomcat-5.5-doc/logging.html>

```
log4j.category.org.some.warn.package=WARN
```

2.1.2. Configuring client side developer logs

Client side logging can be configured in `$GLOBUS_LOCATION/log4j.properties`. The logger `log4j.appender.A1` is used for developer logging and by default writes output to the system output. By default it is set for all warnings in the Globus Toolkit package to be displayed.

2.2. Configuring system administration logs

The specific logger to edit will be `log4j.logger.sysadmin` in `$GLOBUS_LOCATION/container-log4j.properties`. There you can configure the following properties:

```
log4j.appender.infoCategory=org.apache.log4j.RollingFileAppender
log4j.appender.infoCategory.Threshold=INFO
log4j.appender.infoCategory.File=var/containerLog
log4j.appender.infoCategory.MaxFileSize=10MB
log4j.appender.infoCategory.MaxBackupIndex=2
```

Above implies the logging file is rolling with each file size limited to 10MB and the logging information is stored in `$GLOBUS_LOCATION/var/containerLog`.

2.3. Sample log file

The [sample log file](#)⁶ contains many log entries for various scenarios in the Java WS container.

⁶ <http://www.globus.org/toolkit/docs/4.2/4.2.0/common/javawscore/sample-container-log.txt>

Chapter 6. Troubleshooting

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

1. Errors

Table 6.1. Data Replication Service (DRS) Errors

Error Code	Definition	Possible Solutions
Authorization failed. Expected <hostname1> target but received <hostname2>	Did not receive expected hostname	When authorization is enabled on the container, you may need to use the proper hostname when referencing the DRS service rather than using localhost.
org.globus.wsrp.ResourceException: Failed to create Replication: /scratch/testrun (No such file or directory)	Cannot find the request file	Ensure that the request file's filename is correct, that it is reachable by the DRS service, and that it has the appropriate permissions for the DRS service to access it.
org.globus.wsrp.ResourceException: Failed to create Replication: String index out of range: -1	The request file is malformed (for example by using spaces instead of a delimiting tab character) which is resulting in a runtime exception.	Make sure your request file is in the correct form as described here .

Index

D

debugging
 logging, 6

E

errors, 8

I

installing
 using binary, 1
 using CVS, 1
 using source, 1

L

logging, 6
 CEDPS-compliant, 6
 debugging, 6

DRAFT