GT 4.0 Tech Preview: Data Replication Service (DRS)

GT 4.0 Tech Preview: Data Replication Service (DRS)

Table of Contents

List of Tables

Chapter 1. Key Concepts

See the key concepts [1] from Data Management.

[1] http://www.globus.org/toolkit/docs/4.0/data/key

Chapter 2. GT 4.0 Release Notes: Data Replication Service (DRS)

1. Component Overview

The Data Replication Service (DRS) is a *technical preview* provided with the Globus Toolkit 4.0 and first appeared in the GT 3.9.5 Beta release. The primary functionality of the component allows users to identify a set of desired files existing in their Grid environment, to make local replicas of those data files by transferring files from one or more source locations, and to register the new replicas in a Replica Location Service. The DRS conforms to the WS-RF specification and exposes a WS-Resource (called a "Replicator" resource) which represents the current state of the requested replication activity and allows users to query or subscribe to various Resource Properties in order to monitor the state of the resource. The DRS is built on the GT 4.0 Java WS Core and uses the Globus RLS to locate and register replicas and the Globus RFT to transfer files.

2. Feature Summary

Features new in release GT 4.0:

- Initial implementation of the Data Replication Service: a WS-Resource, called the *Replicator*, which accepts a request from a client to locate, transfer, and register new replicas of data files in the Grid environment.
- A command-line tool to create and start (`globus-replication-create`) replication requests.
- WSDL-defined SOAP operation to *create* (and implicitly starts) a replication request. For details, click here[1] to view a listing of the WSDL-defined interface from the Globus CVS repository.

Other Supported Features

- Supports secure transport, secure conversation, and secure message communication as provided by GT 4.0.

Deprecated Features

- None.

3. Technology Dependencies

DRS depends on the following GT components:

- Java WS Core
- WS Authentication and Authorization
- Delegation Service
- RFT
- RLS

DRS depends on the following 3rd party software:

[1] http://viewcvs.globus.org/viewcvs.cgi/ws-replica/replicator/common/schema/replica/replicator/?only_with_tag=globus_4_0_0

- PostgreSQL

4. Supported Platforms

Tested Platforms for DRS

- Linux (RedHat, Debian)

5. Backward Compatibility Summary

Protocol changes since GT version 3.2:

- None

API changes since GT version 3.2:

- None

Exception changes since GT version 3.2:

- None

Schema changes since GT version 3.2:

- None

6. For More Information

Click here[2] for more information about this component.

[2] index.html

Chapter 3. GT 4.0.1 Incremental Release Notes: Data Replication Service (DRS)

1. Introduction

These release notes are for the incremental release 4.0.1. It includes a summary of changes since 4.0.0, bug fixes since 4.0.0 and any known problems that still exist at the time of the 4.0.1 release. This page is in addition to the top-level 4.0.1 release notes at http://www.globus.org/toolkit/releasenotes/4.0.1.

For release notes about 4.0 (including feature summary, technology dependencies, etc) go to the DRS 4.0 Release Notes[1].

2. Changes Summary

As a technical preview release the Data Replication Service (DRS) has undergone numerous changes to its schema, exceptions, and API. We do not expect the interfaces to change as dramatically between this release and the next full release. Changes since the GT 4.0.0 release of DRS include:

- Rewrite of the WSDL and Schema
- Refactoring of the backend implementation into a reusable JNDI resource
- Support for "bulk" operations on the RLS for discovery and registration of replicas
- Concurrent querying of all replica catalogs (LRCs) to locate all available replicas efficiently
- Splitting of RFT transfer requests into configurable-sized chunks to avoid SOAP message size limitations
- Introduction of source selector interfaces to allow user-defined source selection algorithms
- New set of command-line tools to fully exploit all DRS operations as defined by its WSDL
- DRS service now downloads (via http, ftp) remotely-hosted request files to avoid SOAP message size limitations
- Support for fine-grained lifecycle control via start, stop, suspend, and resume operations
- Replicator resource is no longer backed by database persistence to avoid DBMS setup
- Extensize set of configuration options to tune DRS behavior
- Support for all RFT options (e.g., tcp buffer size, parallelism, etc.)

3. Bug Fixes

Significant improvements to the stability, scalability and usability of the DRS service have been made since the initial introduction of the DRS in the GT 4.0.0 release. Detailed tracking of bugs and corresponding fixes will begin with this release.

[1] http://www.globus.org/toolkit/docs/4.0/techpreview/datarep/DataRep_Release_Notes.html

4. Known Problems

The following problem is known to exist for DRS at the time of the 4.0.1 release:

- Bug 3502:[2] Container freezes (CPU spin) when making remote call between ws-resources. See record for *WORKAROUND*.

5. For More Information

Click here[3] for more information about this component.

[2] http://bugzilla.globus.org/globus/show_bug.cgi?id=3502
[3] index.html

Chapter 4. GT 4.0.2 Incremental Release Notes: Data Replication Service (DRS)

1. Introduction

These release notes are for the incremental release 4.0.2. It includes a summary of changes since 4.0.1, bug fixes since 4.0.1 and any known problems that still exist at the time of the 4.0.2 release. This page is in addition to the top-level 4.0.2 release notes at http://www.globus.org/toolkit/releasenotes/4.0.2.

For release notes about 4.0 (including feature summary, technology dependencies, etc) go to the DRS 4.0 Release Notes[1].

2. Changes Summary

As indicated in the release notes for the 4.0.1 release of the DRS, we have stabilized the interfaces to the service. The WSDL remains unchanged from the 4.0.1 release. The bug fixes for this release significantly improve the scalability of the service. We have tested concurrent operations and scalability and resolved problems in the DRS code as well as in its dependent components (Java WS Core and RFT) which affected DRS performance.

3. Bug Fixes

- Bug 4202:[2] Replication requests fail or perpetually pending due to erroneous RFT status code
- Bug 4203:[3] Internal replication operations performed synchronously due to WorkManager bug
- Bug 4224:[4] DRS misinterprets RFT failure status
- Bug 4230:[5] DRS deadlocks in thread scheduling

4. Known Problems

The following problems are known to exist for DRS at the time of the 4.0.2 release:

- See Java WS Core buglist concerning container freeze, which may affect DRS performance: Bug 3502:[6] Container freezes (CPU spin) when making remote call between ws-resources. See record for *WORKAROUND*.
- Bug 4231:[7] DRS does not implement subscription/notification

[1] http://www.globus.org/toolkit/docs/4.0/techpreview/datarep/DataRep_Release_Notes.html
[2] http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=4202
[3] http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=4203
[4] http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=4224
[5] http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=4230
[6] http://bugzilla.globus.org/globus/show_bug.cgi?id=3502
[7] http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=4231

5. For More Information

Click here[8] for more information about this component.

[8] index.html

Chapter 5. GT 4.0.3 Incremental Release Notes: Data Replication Service (DRS)

1. Introduction

These release notes are for the incremental release 4.0.3. It includes a summary of changes since 4.0.2, bug fixes since 4.0.2 and any known problems that still exist at the time of the 4.0.3 release. This page is in addition to the top-level 4.0.3 release notes at http://www.globus.org/toolkit/releasenotes/4.0.3.

For release notes about 4.0 (including feature summary, technology dependencies, etc) go to the DRS 4.0 Release Notes[1].

2. Changes Summary

No changes have occurred since GT 4.0.2.

3. Bug Fixes

There are no fixed bugs GT 4.0.2.

4. Known Problems

The following problems are known to exist for DRS at the time of the 4.0.3 release.

- See Java WS Core buglist concerning container freeze, which may affect DRS performance: Bug 3502:[2] Container freezes (CPU spin) when making remote call between ws-resources. See record for *WORKAROUND*.
- Bug 4231:[3] DRS does not implement subscription/notification

5. For More Information

Click here[4] for more information about this component.

[1] http://www.globus.org/toolkit/docs/4.0/techpreview/datarep/DataRep_Release_Notes.html
[2] http://bugzilla.globus.org/globus/show_bug.cgi?id=3502
[3] http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=4231
[4] index.html

Chapter 6. GT 4.0.4 Incremental Release Notes: Data Replication Service (DRS)

1. Introduction

These release notes are for the incremental release 4.0.4. It includes a summary of changes since 4.0.3, bug fixes since 4.0.3 and any known problems that still exist at the time of the 4.0.4 release. This page is in addition to the top-level 4.0.4 release notes at http://www.globus.org/toolkit/releasenotes/4.0.4.

For release notes about 4.0 (including feature summary, technology dependencies, etc) go to the DRS 4.0 Release Notes[1].

2. Changes Summary

No changes have occurred since GT 4.0.3.

3. Bug Fixes

There are no new bug fixes.

4. Known Problems

- See Java WS Core buglist concerning container freeze, which may affect DRS performance: Bug 3502:[2] Container freezes (CPU spin) when making remote call between ws-resources. See record for *WORKAROUND.*
- Bug 4231:[3] DRS does not implement subscription/notification

5. For More Information

Click here[4] for more information about this component.

[1] http://www.globus.org/toolkit/docs/4.0/techpreview/datarep/DataRep_Release_Notes.html
[2] http://bugzilla.globus.org/globus/show_bug.cgi?id=3502
[3] http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=4231
[4] index.html

Chapter 7. GT 4.0 DRS: System Administrator's Guide

1. 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 GT 4.0 System Administrator's Guide[1]. Read through this guide before continuing!

2. Building and installing

DRS is not installed as part of the standard install. In order to install DRS, you will need to ensure that RLS is also installed as part of the build by setting `GLOBUS_IODBC_PATH` or configuring with the `--with-iodbc` flag. Also, the `--enable-drs` flag is required as part of the configuration step. The following commands will install the DRS.

```
% ./configure --prefix=/path/to/install --with-iodbc=/path/to/iodbcdir --enable-drs
% make
% make install
```

Important

The DRS depends on the Globus RLS. In order for the "make" command above to build and install the RLS, you must follow the basic installation instructions for GT 4.0 as noted previously. Installation of the RLS requires that the `--with-iodbc` configure flag or the `GLOBUS_IODBC_PATH` environment variable be properly set *before* issuing the `make` command. Not setting this environment variable before installation is a common mistake encountered by users.

Additionally, you may access the 4.0.1 branch of DRS from the Globus Toolkit CVS repository. You may deploy the 4.0.1 DRS on top of an existing GT 4.0.0 installation as well as a GT 4.0.1 installation. The following instructions will only work given the presence of an existing GT 4.0.0 or GT 4.0.1 installation with *all DRS dependencies*. The best way to establish such an environment is to first follow the above instructions and subsequently use the following instructions to obtain the latest updates from the CVS branch.

```
% setenv GLOBUS_LOCATION /path/to/existing/gt40x/install/
% setenv CVSROOT :pserver:anonymous@cvs.globus.org:/home/globdev/CVS/globus-packages
% cvs co -r globus_4_0_branch ws-replica/
% cd ws-replica/replicator/
% ant deploy
```

To learn more about general instructions regarding GT CVS access, see Remote CVS Access[2].

[1] ../../admin/docbook/
[2] http://www.globus.org/toolkit/docs/development/remote-cvs.html

3. Configuring

This information is in addition to the basic configuration instructions in the GT4.0 System Administrator's Guide[3]. Aside from the basic configuration of GT 4.0, please review the following instructions:

3.1. 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_wsrf_replicator/jndi-config.xml`. To view the default configuration file (shipped with the GT 4.0.1 release) from the Globus CVS repository click here[4].

3.2. 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.

4. Deploying

No further information is necessary.

[3] http://www.globus.org/toolkit/docs/4.0/admin/docbook/
[4] http://viewcvs.globus.org/viewcvs.cgi/ws-replica/replicator/service/java/source/deploy-jndi-config.xml?tab=globus_4_0_branch&content-type=text/vnd.viewcvs-markup

5. Testing

This service does not provide a set of tests yet.

6. Security considerations

6.1. 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.

6.2. 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.

7. Troubleshooting

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
...
```

Chapter 8. GT 4.0 DRS: User's Guide

1. Introduction

The Data Replication Service (DRS) is a *technical preview* provided with the Globus Toolkit 4.0 and first appeared in the GT 3.9.5 Beta release. The primary functionality of the component allows users to identify a set of desired files existing in their Grid environment, to make local replicas of those data files by transferring files from one or more source locations, and to register the new replicas in a Replica Location Service. The DRS conforms to the WS-RF specification and exposes a WS-Resource (called a "Replicator" resource) which represents the current state of the requested replication activity and allows users to query or subscribe to various Resource Properties in order to monitor the state of the resource. The DRS is built on the GT 4.0 Java WS Core and uses the Globus RLS to locate and register replicas and the Globus RFT to transfer files.

2. Command-line tools

Please see the GT 4.0 DRS Command-line Reference.

3. Usage scenarios

This section describes a few key usage scenarios and provides examples of using the DRS command-line tools.

3.1. Generate a valid proxy

Before using any of the tools, a user must generate a valid user proxy. Use grid-proxy-init[1].

```
% $GLOBUS_LOCATION/bin/grid-proxy-init
Your identity: /O=Grid/OU=GlobusTest/OU=simpleCA.mymachine/OU=mymachine/CN=John Doe
Enter GRID pass phrase for this identity:
Creating proxy ................................ Done
Your proxy is valid until: Tue Oct 26 01:33:42 2004
```

3.2. Delegate user credentials

Once you have generated a valid proxy you must create a delegated credential. Your delegated credential will be used by the DRS to make secure calls to other services (e.g., RLS, RFT, etc.) in order to perform the data replication. It is important to ensure that you give your delegated credential enough `lifetime` to support the running time of your replication activities. To delegate your credential use globus-credential-delegate[2].

```
% $GLOBUS_LOCATION/bin/globus-credential-delegate -h myhostname \
 -p 8443 mycredential.epr
EPR will be written to: mycredential.epr
Delegated credential EPR:
Address: https://128.9.72.118:8443/wsrf/services/DelegationService
Reference property[0]:
```

[1] http://www.globus.org/toolkit/docs/4.0/security/prewsaa/rn01re05.html
[2] http://www.globus.org/toolkit/docs/4.0/security/delegation/rn01re01.html

```
<ns1:DelegationKey xmlns:ns1="http://www.globus.org/08/2004/delegationService"
>3b6cb210-e9b2-11d9-ab74-f7fa10f094cd</ns1:DelegationKey>
```

3.3. Replication request file

A key parameter for any replication request is the request file. The replication request file is a text file containing CRLF-terminated rows of tab-delimited pairs of Logical Filename (LFN) names and destination (URL) locations. An example of such a file is shown.

```
% cat testrun.req
testrun-1      gsiftp://myhost:9001/sandbox/files/testrun-1
testrun-2      gsiftp://myhost:9001/sandbox/files/testrun-2
testrun-3      gsiftp://myhost:9001/sandbox/files/testrun-3
testrun-4      gsiftp://myhost:9001/sandbox/files/testrun-4
testrun-5      gsiftp://myhost:9001/sandbox/files/testrun-5
```

3.4. Create replication resource

The initial step for any replication is to create the replication resource. Creating the resource depends on the availability of a DRS service, a delegated credential, and a properly formatted replication request file. The replication request file must be specified by its URL. Currently supported URL schemes for the request file include `file`, `http`, and `ftp`. If the replication client is run local to the service the `file` scheme is appropriate, whereas if the client is remote than the latter schemes must be used. It is a good practice to specify a filename to save the replication resource's endpoint reference. The endpoint reference is required for all other operations on the resource, such as getting resource properties, starting/stopping, and destroying it. Numerous options are available to influence the behavior of the data replication activities (see globus-replication-create(1)). One option of particular interest is the `--start` option, which immediately starts the replication activities following creation of the replication resource. An example of using the globus-replication-create(1) tool is shown.

```
% $GLOBUS_LOCATION/bin/globus-replication-create -s \
 https://myhost:8443/wsrf/services/ReplicationService \
 -C mycredential.epr -V myreplicator.epr file:///scratch/testrun.req
```

This command does not write to `stdout` when successful unless the `--debug` option is specified.

3.5. Start replication

Once a replication resource has been create, the replication activities may be started. As mentioned in Section 3.4, “Create replication resource” the replication may be immediately started after it is created. If the immediate start option is not specified, the globus-replication-start(1) tool must be used to start the replication.

```
% $GLOBUS_LOCATION/bin/globus-replication-start -e myreplicator.epr
```

No output is expect from this command when successful.

3.6. Get replication resource properties

Throughout the lifecycle and after the completion of the replication resource, it will be important to lookup its Resource Properties. The standard WS-RF port types are supported and the supplied tools (e.g., wsrf-get-property[3]) may be used with the DRS and its resources.

```
% $GLOBUS_LOCATION/bin/wsrf-get-property -e myreplicator.epr \
 "{http://www.globus.org/namespaces/2005/05/replica/replicator}status"
<ns1:status xmlns:ns1="http://www.globus.org/namespaces/2005/05/replica/replicator">
Active</ns1:status>
```

And,

```
% $GLOBUS_LOCATION/bin/wsrf-get-property -e myreplicator.epr \
 "{http://www.globus.org/namespaces/2005/05/replica/replicator}count"
<ns1:count xmlns:ns1="http://www.globus.org/namespaces/2005/05/replica/replicator">
 <ns1:total>10</ns1:total>
 <ns1:finished>0</ns1:finished>
 <ns1:failed>0</ns1:failed>
 <ns1:terminated>0</ns1:terminated>
</ns1:count>
```

3.7. Find replication item status

Throughout the lifecycle and after the completion of the replication resource, it may be helpful to find individual replication items in the replication resource to inspect the detailed status of the replication activities. The globus-replication-finditems(1) tool is used to find replication items. The following example demonstrates the usage when finding a limited number of items, offset into the lookup results set, for a specified status.

```
% $GLOBUS_LOCATION/bin/globus-replication-finditems -e myreplicator.epr -S Pending -O 1 -L
<ns1:FindItemsResponse xmlns:ns1="http://www.globus.org/namespaces/2005/05/replica/replica
 <ns1:items xsi:type="ns1:ReplicationItemType" xmlns:xsi="http://www.w3.org/2001/XMLSchema
  <ns1:uri xsi:type="xsd:string" xmlns:xsd="http://www.w3.org/2001/XMLSchema">testrun-2</n
  <ns1:priority xsi:type="xsd:int" xmlns:xsd="http://www.w3.org/2001/XMLSchema">1000</ns1:
  <ns1:status xsi:type="ns1:ReplicationItemStatusEnumerationType">Pending</ns1:status>
  <ns1:destinations xsi:type="ns1:DestinationType">
   <ns1:uri xsi:type="xsd:string" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
     gsiftp://myhost:9001/sandbox/files/testrun-2</ns1:uri>
   <ns1:status xsi:type="ns1:DestinationStatusEnumerationType">Pending</ns1:status>
  </ns1:destinations>
 </ns1:items>
 <ns1:items xsi:type="ns1:ReplicationItemType" xmlns:xsi="http://www.w3.org/2001/XMLSchema
  <ns1:uri xsi:type="xsd:string" xmlns:xsd="http://www.w3.org/2001/XMLSchema">testrun-3</n
  <ns1:priority xsi:type="xsd:int" xmlns:xsd="http://www.w3.org/2001/XMLSchema">1000</ns1:
  <ns1:status xsi:type="ns1:ReplicationItemStatusEnumerationType">Pending</ns1:status>
  <ns1:destinations xsi:type="ns1:DestinationType">
```

[3] http://www.globus.org/toolkit/docs/4.0/common/javawscore/rn01re08.html

```
   <ns1:uri xsi:type="xsd:string" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
     gsiftp://myhost:9001/sandbox/files/testrun-3</ns1:uri>
   <ns1:status xsi:type="ns1:DestinationStatusEnumerationType">Pending</ns1:status>
  </ns1:destinations>
 </ns1:items>
</ns1:FindItemsResponse>
```

3.8. Destroy replication resource

When the replication is complete, the replication resource may be destroyed. Destroying the replication resource will help to free up system resources (namely, memory), especially in the case that the replication entailed a large amount of individual replication activities (i.e., many files specified in the replication request file). The standard WS-RF port types are supported and the supplied wsrf-destroy[4] tool may be used to destroy the DRS resource.

```
% $GLOBUS_LOCATION/bin/wsrf-destroy -e myreplicator.epr
Destroy operation was successful
```

4. Troubleshooting

The following section provides information about common troubleshooting tips for end users.

4.1. Authorization failure: 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*.

```
% $GLOBUS_LOCATION/bin/globus-replication-create -s \
 https://localhost:8443/wsrf/services/ReplicationService \
 -C mycredential.epr -V myreplicator.epr file:///scratch/testrun.req
Error: ; nested exception is:
        org.globus.common.ChainedIOException: Authentication failed [Caused by:
        Operation unauthorized (Mechanism level: Authorization failed. Expected
        "/CN=host/loopback" target but received "/C=US/O=Globus Alliance/OU=
        Service/CN=host/myhost")]
```

4.2. Cannot find request file

When using the DRS, 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.

```
% $GLOBUS_LOCATION/bin/globus-replication-create -s \
 https://myhost:8443/wsrf/services/ReplicationService -C mycredential.epr \
 -V myreplicator.epr file:///scratch/testrun
Error: java.rmi.RemoteException: Unable to create resource; nested exception is:
```

[4] http://www.globus.org/toolkit/docs/4.0/common/javawscore/rn01re05.html

```
        org.globus.wsrf.ResourceException: Failed to create Replication:
        /scratch/testrun (No such file or directory); nested exception is:
        java.io.FileNotFoundException: /scratch/testrun (No such file or directory)
```

4.3. Malformed request file

It is important to ensure that the request file is well-formed as specified. A malformed request file will result in a runtime exception.

```
% $GLOBUS_LOCATION/bin/globus-replication-create -s \
 https://myhost:8443/wsrf/services/ReplicationService -C mycredential.epr \
 -V myreplicator.epr file:///scratch/testrun.req
Error: java.rmi.RemoteException: Unable to create resource; nested exception is:
        org.globus.wsrf.ResourceException: Failed to create Replication: String
        index out of range: -1; nested exception is:
        java.lang.StringIndexOutOfBoundsException: String index out of range: -1
```

The above error was produced by replacing a delimiting `tab` character with `space` characters.

Chapter 9. GT 4.0 DRS: Developer's Guide

1. Introduction

This guide contains information of interest to developers working with DRS. It provides reference information for application developers, including APIs, architecture, procedures for using the APIs and code samples.

2. Before you begin

2.1. Feature summary

Features new in release GT 4.0.1:

- Improved implementation of the Data Replication Service: a WS-Resource, called the *Replicator*, which accepts a request from a client to locate, transfer, and register new replicas of data files in the Grid environment.
- A set of command-line tools to create (`globus-replication-create`), start (`globus-replication-start`), stop (`globus-replication-stop`), suspend (`globus-replication-suspend`), resume (`globus-replication-resume`) replication requests, and find item status (`globus-replication-finditems`).
- WSDL-defined SOAP operations to *create*, *start*, *stop*, *suspsend*, and *resume* a replication request, along with operations to get the status of individual replicas in the request. For details, click here[1] to view a listing of the WSDL-defined interface from the Globus CVS repository.
- APIs to allows users to implement custom replica source selection algorithms.

Other Supported Features

- Supports secure transport, secure conversation, and secure message communication as provided by GT 4.0.

Deprecated Features

- Database-backed State Persistence: State is now maintained in memory and lasts only for the lifetime of the WS-Resource or as dictated by the service container. This change simplifies setup of the DRS. We intend to reintroduce other persistence model(s) after we have collected additional user feedback on the DRS.

2.2. Tested platforms

Tested Platforms for DRS

- Linux (RedHat, Debian)

2.3. Backward compatibility summary

Protocol changes since GT version 3.2:

[1] http://viewcvs.globus.org/viewcvs.cgi/ws-replica/replicator/common/schema/replica/replicator/?only_with_tag=globus_4_0_branch

- None

API changes since GT version 3.2:

- None

Exception changes since GT version 3.2:

- None

Schema changes since GT version 3.2:

- None

2.4. Technology dependencies

DRS depends on the following GT components:

- Java WS Core
- WS Authentication and Authorization
- Delegation Service
- RFT
- RLS

DRS depends on the following 3rd party software:

- None

2.5. Security considerations

2.5.1. 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.

2.5.2. 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.

3. Architecture and design overview

For a review of the DRS architecture and design please see Wide Area Data Replication for Scientific Collaboration[2].

4. Public interface

The semantics and syntax of the APIs for this component can be found in the public interface guide[3].

5. Usage scenarios

Not available.

6. Debugging

Not available.

7. Troubleshooting

Not available.

8. Related Documentation

Not available.

[2] http://www.isi.edu/~annc/papers/grid2005submitted.pdf
[3] DataRep_Public_Interfaces.html

Chapter 10. GT 4.0 Component Fact Sheet: Data Replication Service (DRS)

1. Brief component overview

The Data Replication Service (DRS) is a *technical preview* provided with the Globus Toolkit 4.0 and first appeared in the GT 3.9.5 Beta release. The primary functionality of the component allows users to identify a set of desired files existing in their Grid environment, to make local replicas of those data files by transferring files from one or more source locations, and to register the new replicas in a Replica Location Service. The DRS conforms to the WS-RF specification and exposes a WS-Resource (called a "Replicator" resource) which represents the current state of the requested replication activity and allows users to query or subscribe to various Resource Properties in order to monitor the state of the resource. The DRS is built on the GT 4.0 Java WS Core and uses the Globus RLS to locate and register replicas and the Globus RFT to transfer files.

2. Summary of features

Features new in release GT 4.0.1:

- Improved implementation of the Data Replication Service: a WS-Resource, called the *Replicator*, which accepts a request from a client to locate, transfer, and register new replicas of data files in the Grid environment.
- A set of command-line tools to create (`globus-replication-create`), start (`globus-replication-start`), stop (`globus-replication-stop`), suspend (`globus-replication-suspend`), resume (`globus-replication-resume`) replication requests, and find item status (`globus-replication-finditems`).
- WSDL-defined SOAP operations to *create*, *start*, *stop*, *suspsend*, and *resume* a replication request, along with operations to get the status of individual replicas in the request. For details, click here[1] to view a listing of the WSDL-defined interface from the Globus CVS repository.
- APIs to allows users to implement custom replica source selection algorithms.

Other Supported Features

- Supports secure transport, secure conversation, and secure message communication as provided by GT 4.0.

Deprecated Features

- Database-backed State Persistence: State is now maintained in memory and lasts only for the lifetime of the WS-Resource or as dictated by the service container. This change simplifies setup of the DRS. We intend to reintroduce other persistence model(s) after we have collected additional user feedback on the DRS.

3. Backward compatibility summary

Protocol changes since GT version 3.2:

- None

[1] http://viewcvs.globus.org/viewcvs.cgi/ws-replica/replicator/common/schema/replica/replicator/?only_with_tag=globus_4_0_branch

API changes since GT version 3.2:

- None

Exception changes since GT version 3.2:

- None

Schema changes since GT version 3.2:

- None

4. Technology dependencies

DRS depends on the following GT components:

- Java WS Core
- WS Authentication and Authorization
- Delegation Service
- RFT
- RLS

DRS depends on the following 3rd party software:

- None

5. Tested platforms

Tested Platforms for DRS

- Linux (RedHat, Debian)

6. Associated standards

Associated standards for DataRep:

- WS-RF[2]
- WS-Addressing[3]
- WS-Security[4]

7. For More Information

Click here[5] for more information about this component.

[2] http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-02.pdf
[3] http://msdn.microsoft.com/ws/2004/03/ws-addressing
[4] http://msdn.microsoft.com/webservices/understanding/specs/default.aspx?pull=/library/en-us/dnglobspec/html/wssecurspecindex.asp
[5] index.html

Chapter 11. GT 4.0 Component Guide to Public Interfaces: Data Replication Service (DRS)

1. Semantics and syntax of APIs

1.1. Programming Model Overview

The DRS is a WS-RF compliant service implemented using the Globus Java WS Core. It exposes a set of Resource Properties and operations to allow users to create replication resources, control replication resources' lifecycle, and inspect the state of replication resources' activities along with the success or failure of individual replicated data sets. In this release, the WSDL and the command-line clients are the primary public interfaces for developers. Two java interfaces exist on the service-side to allow developers and users to modify the source selection behavior of the DRS. These interfaces allow users to chose alternate schemes to select sources beyond the random selection provided by default.

1.2. Component API

Interfaces to influence source selection include:

- ReplicaCatalogFilter
- SourceSelector

Please see service-side interfaces[1] for documentation on these interfaces.

2. Semantics and syntax of the WSDL

2.1. Protocol overview

The DRS provides a set of Resource Properties and SOAP operations to create, manipulate and inspect replication activities. Users will begin by creating a replication resource (AKA "Replicator") by invoking the create operation and passing it a URL of the replication request file (described in the domain-specific interface section). Users may start, stop, suspend and resume the Replicator when necessary. Typically a user is expected to simply start the resource and allow it to run through completion. During and after the course of replication activities performed by the resource, users may invoke standard "get resource property" and DRS-specific "find" operations to inspect the state of the resource. When the resource finishes the replication activities and the user has satisfactorily inspected the resource state, the resource should be destroyed using the standard "destroy" operation.

2.2. Operations

Supported operations include:

- `createReplicator` creates the "Replicator" resource.

[1] http://www.isi.edu/~schuler/drsdocs401/

- `[in] InitialTerminationTime` The requested initial termination time for the resource.
- `[in] requestFileRequest` The request-file style request.
  - `credentialEPR` Endpoint Reference of the user's delegated credential.
  - `options` Replication options which include a set of options pertinent to the transfer stage of the request, such as concurrency, parallel streams, tcp buffer size, etc.
  - `autostart` A Boolean flag indicating whether the resource should be automatically started following resource creation.
  - `requestFileUri` The URI of the request file. Currently supported schemes include http, file, and ftp.
  - `format` The request file format (domain-specific). Currently, the service only supports a simple "Table" format.
- `[out] EPR` The Endpoint Reference of the Replicator resource.
- `[fault] fault` Indicates a general failure when attempting to create the Replicator resource.

- `start` starts the resource.
  - `[fault] invalidStateFault` Indicates the resource is in an invalid state to perform the operation.
- `stop` stops the resource.
  - `[fault] invalidStateFault` Indicates the resource is in an invalid state to perform the operation.
- `suspend` suspends the resource.
  - `[fault] invalidStateFault` Indicates the resource is in an invalid state to perform the operation.
- `resume` resumes the resource.
  - `[fault] invalidStateFault` Indicates the resource is in an invalid state to perform the operation.
- `findItems` Finds state information for individual replication items.
  - `[in] byUri` Finds by replication URI (currently, this value must be the logical filename, LFN, rather than a properly formed URI). This param is mutually-exclusive with `byStatus`.
  - `[in] byStatus` Find by status, which includes `Pending`, `Finished`, `Failed`, and `Terminated`. This param is mutually-exclusive with `byUri`.
  - `[in] offset` An offset into the results set.
  - `[in] limit` A limit of results to be returned to the client.
  - `[out] items` An array of items to be returned to the client as a result of the find operation. Each item in the array contains the complete status of the replication item including its identifier, priority, status, error (if any), sources, and destinations.
  - `[fault] internalErrorFault` Indicates that an internal error occurred.

2.3. Resource properties

Supported resource properties for DataRep include:

- `status`: The status of the resource, such as Pending, Active, Suspended, Terminated, Destroyed, etc.
- `stage`: The current stage or activity of the resource, such as Discover, Transfer, and Register.
- `result`: The final result (if any) of the resource, such as Finished, Failed, and Exception.
- `errorMessage`: A verbose description of an error (if any) encountered by the resource. The message may include error or exception information returned by one of the dependent components, such as RLS or RFT.
- `count`: An element containing counts of individual replication items pertaining to total, finished, failed, and terminated replication items.

2.4. Faults

Supported faults include:

- `CreateReplicatorFault` Indicates that the service failed to create the Replicator resource.
- `RequestBodyMissingFault` Indicates that the request body of the create message parameters was missing.
- `CredentialEprMissingFault` Indicates that the delegated credential EPR was missing from the create message.
- `InvalidStateFault` Indicates that the requested lifecycle operation (e.g., start, stop, suspend, resume) was performed on a resource that was not in the proper state for the operation to succeed (e.g., performing a resume operation on a non-suspended Replicator resource).
- `InternalErrorFaultType` Indicates that an internal error occurred (e.g., internal system failure, etc.).

2.5. WSDL and Schema Definition

For more information, please see the Replicator Port Type[2] or the complete list of schemas[3].

3. Command-line tools

Please see the GT 4.0 DRS Command-line Reference.

4. Overview of Graphical User Interface

4.1. Overview of the purpose and functionality of the GUI

The DRS does not provide a GUI interface.

[2] http://viewcvs.globus.org/viewcvs.cgi/ws-replica/replicator/common/schema/replica/replicator/replicator_port_type.wsdl?rev=1.2.2.1&only_with_tag=globus_4_0_branch&content-type=text/vnd.viewcvs-markup
[3] http://viewcvs.globus.org/viewcvs.cgi/ws-replica/replicator/common/schema/replica/replicator/?only_with_tag=globus_4_0_branch

4.2. Command and options

Not available

4.3. Limitations

Not available

5. Semantics and syntax of domain-specific interface

5.1. Interface introduction

The DRS domain-specific interface defines the structure and expected contents of a request file used when creating a replication resource. When the client invokes the create operation of the DRS, it will be expected to submit a properly formatted request file. It is important to understand the structure of the request file and to ensure that the file is well-formed.

5.2. Syntax of the interface

For the present release, the DRS request file format is fairly trivial. The request file is structured as a "Table" style of rows and columns of text. Each row represent a requested replication item described in two columns. The first column contains the identifier of the data set which will be discovered and replicated. The identifier must be resolvable by the Replica Location Index (see the JNDI configuration for `defaultIndexUrl`). In most cases, it is expected that the identifier be a Logical Filename (LFN) per the Replica Location Service definition. The second column of the row contains the URL of the "destination" for the replication item. The two columns must be delimited by a TAB character and each row must be delimited by an EOL character.

Note

The service will not accept SPACE characters as a substitute for the TAB character.

The following example shows the output of a small request file.

```
% cat example.req
my-lfn-1        gsiftp://myhost:9001/sandbox/examples/files/my-pfn-1
my-lfn-2        gsiftp://myhost:9001/sandbox/examples/files/my-pfn-2
my-lfn-3        gsiftp://myhost:9001/sandbox/examples/files/my-pfn-3
my-lfn-4        gsiftp://myhost:9001/sandbox/examples/files/my-pfn-4
my-lfn-5        gsiftp://myhost:9001/sandbox/examples/files/my-pfn-5
```

6. Configuration interface

Please see the GT 4.0 DRS Command-line Reference.

7. Environment variable interface

- `GLOBUS_LOCATION`=*/path/to/globus/install*

Chapter 12. GT 4.0 DRS: Quality Profile

1. Test coverage reports

Not available.

2. Code analysis reports

Not available.

3. Outstanding bugs

See a bugzilla[1] query on `Replication Services`, `DRS` to list the bugs outstanding.

4. Bug Fixes

Significant improvements to the stability, scalability and usability of the DRS service have been made since the initial introduction of the DRS in the GT 4.0.0 release. Detailed tracking of bugs and corresponding fixes will begin with this release.

5. Performance reports

For a review of the DRS performance, please see Wide Area Data Replication for Scientific Collaboration[2].

[1] http://bugzilla.globus.org/globus/query.cgi
[2] http://www.isi.edu/~annc/papers/grid2005submitted.pdf

Chapter 13. GT 4.0 Migrating Guide for Data Replication Service (DRS)

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

1. Migrating from GT2

This component is new with GT version 4.0.0; therefore, this section does not apply.

2. Migrating from GT3

This component is new with GT version 4.0.0; therefore, this section does not apply.

GT 4.0: Data Replication Service Command Reference

The DRS provides a set of command-line tools to control the creation and lifecycle of a given replication request. These command line tools are available on Unix and Windows platforms and will work in the same way (of course within the platform rules - the path syntax, variable definitions, etc.).

Name

globus-replication-create -- This tool is used to create a replication resource by submitting a replication request to the designated replication service.

`globus-replication-create`

Tool description

Use this tool to create replication resources (also refered to as "Replicator" resources). You must specify the URL of the ReplicationService where the resource will be created. You must submit the filename of a file containing an Endpoint Reference (EPR) to a delegated credential resource, which you must have previously created. Finally, you must submit the URL of a request file specifying the desired data replications. If the client is running local to the service container the URL may be a `file://` URL, whereas if the client is remote the URL may be a `http://` or `ftp://` URL. The request file adopts a table format structure where each line in the file represents a source-destination pair delimited by a single *tab* character. The source should be a logical filename (LFN) as found in a Replica Location Service (RLS) Replica Location Index (RLI) service. The destination should be a URL acceptable to the GridFTP server. Most likely, you will want to specify a filename in order to save the newly created Replicator resource's EPR. You may use the EPR for starting the resource and querying its resource properties.

Command syntax

```
globus-replication-create [options] request-file
```

Table 1. Options

-a,--anonymous	Use anonymous authentication. (requires either -m 'conv' or transport (https) security)
--binary <boolean>	Specifies binary data transfer
--blockSize <int>	Block size for data transfer
-c,--serverCertificate <file>	A file with server's certificate used for encryption. Used in the case of GSI Secure Message encryption
-C,--delegatedCredential <file>	Loads Delegated Credential EPR from file
--concurrency <int>	Concurrency of data transfer
-d,--debug	Enables debug mode
--dataChannelAuth <boolean>	Data channel authentication for transfers
--destinationSubject <name>	Destination subject name for data transfer
-e,--eprFile <file>	Loads EPR from file
-f,--descriptor <file>	Sets client security descriptor. Overrides all other security settings
-g,--delegation <mode>	Performs delegation. Can be 'limited' or 'full'. (requires -m 'conv')
-h,--help	Displays help
-k,--key <name value>	Resource Key
-l,--contextLifetime <value>	Lifetime of context created for GSI Secure Conversation (requires -m 'conv')
-m,--securityMech <type>	Sets authentication mechanism: 'msg' (for GSI Secure Message), or 'conv' (for GSI Secure Conversation)
-p,--protection <type>	Sets protection level, can be 'sig' (for signature) can be 'enc' (for encryption)
--parallelStreams <int>	Parallel streams for data transfer
-s,--service <url>	Service URL
-S,--start	Starts the Replicator resource immediately
--sourceSubject <name>	Source subject name for data transfer
--subject <name>	Subject name for data transfer
--tcpBufferSize <int>	TCP buffer size for data transfer
--userName <name>	User name for data transfer
-V,--saveEpr <file>	Save EPR of newly created Replicator to file
-z,--authorization <type>	Sets authorization, can be 'self', 'host' or 'none'

Name

globus-replication-start -- This tool starts the replication activities.

`globus-replication-start`

Tool description

Replication resources created with the `globus-replication-create` tool may be "started" by using this tool and passing the filename of the saved EPR as a parameter to the tool. The tool will indicate an error condition if the user attempts to start a resource that has been previously started.

Command syntax

```
globus-replication-start [options]
```

Table 2. Options

-a,--anonymous	Use anonymous authentication. (requires either -m 'conv' or transport (https) security)
-c,--serverCertificate <file>	A file with server's certificate used for encryption. Used in the case of GSI Secure Message encryption
-d,--debug	Enables debug mode
-e,--eprFile <file>	Loads EPR from file
-f,--descriptor <file>	Sets client security descriptor. Overrides all other security settings
-g,--delegation <mode>	Performs delegation. Can be 'limited' or 'full'. (requires -m 'conv')
-h,--help	Displays help
-k,--key <name value>	Resource Key
-l,--contextLifetime <value>	Lifetime of context created for GSI Secure Conversation (requires -m 'conv')
-m,--securityMech <type>	Sets authentication mechanism: 'msg' (for GSI Secure Message), or 'conv' (for GSI Secure Conversation)
-p,--protection <type>	Sets protection level, can be 'sig' (for signature) can be 'enc' (for encryption)
-s,--service <url>	Service URL
-z,--authorization <type>	Sets authorization, can be 'self', 'host' or 'none'

Name

globus-replication-stop -- This tool stops the replication activities.

`globus-replication-stop`

Tool description

Replication resources created with the `globus-replication-create` tool may be "stoped" by using this tool and passing the filename of the saved EPR as a parameter to the tool. The tool will indicate an error condition if the user attempts to stop a resource that has not been previously started, a resource that has been suspended, or a resource that has terminated or been destroyed.

Command syntax

```
globus-replication-stop [options]
```

Table 3. Options

-a,--anonymous	Use anonymous authentication. (requires either -m 'conv' or transport (https) security)
-c,--serverCertificate <file>	A file with server's certificate used for encryption. Used in the case of GSI Secure Message encryption
-d,--debug	Enables debug mode
-e,--eprFile <file>	Loads EPR from file
-f,--descriptor <file>	Sets client security descriptor. Overrides all other security settings
-g,--delegation <mode>	Performs delegation. Can be 'limited' or 'full'. (requires -m 'conv')
-h,--help	Displays help
-k,--key <name value>	Resource Key
-l,--contextLifetime <value>	Lifetime of context created for GSI Secure Conversation (requires -m 'conv')
-m,--securityMech <type>	Sets authentication mechanism: 'msg' (for GSI Secure Message), or 'conv' (for GSI Secure Conversation)
-p,--protection <type>	Sets protection level, can be 'sig' (for signature) can be 'enc' (for encryption)
-s,--service <url>	Service URL
-z,--authorization <type>	Sets authorization, can be 'self', 'host' or 'none'

Name

globus-replication-suspend -- This tool suspends the replication activities.

`globus-replication-suspend`

Tool description

Replication resources created with the `globus-replication-create` tool may be "suspended" by using this tool and passing the filename of the saved EPR as a parameter to the tool. The tool will indicate an error condition if the user attempts to suspend a resource that has not been previously started, a resource that has been suspended, or a resources that is done or has been destroyed.

Command syntax

```
globus-replication-suspend [options]
```

Table 4. Options

-a,--anonymous	Use anonymous authentication. (requires either -m 'conv' or transport (https) security)
-c,--serverCertificate <file>	A file with server's certificate used for encryption. Used in the case of GSI Secure Message encryption
-d,--debug	Enables debug mode
-e,--eprFile <file>	Loads EPR from file
-f,--descriptor <file>	Sets client security descriptor. Overrides all other security settings
-g,--delegation <mode>	Performs delegation. Can be 'limited' or 'full'. (requires -m 'conv')
-h,--help	Displays help
-k,--key <name value>	Resource Key
-l,--contextLifetime <value>	Lifetime of context created for GSI Secure Conversation (requires -m 'conv')
-m,--securityMech <type>	Sets authentication mechanism: 'msg' (for GSI Secure Message), or 'conv' (for GSI Secure Conversation)
-p,--protection <type>	Sets protection level, can be 'sig' (for signature) can be 'enc' (for encryption)
-s,--service <url>	Service URL
-z,--authorization <type>	Sets authorization, can be 'self', 'host' or 'none'

Name

globus-replication-resume -- This tool resumes the replication activities.

`globus-replication-resume`

Tool description

Replication resources created with the `globus-replication-create` tool may be "resumed" by using this tool and passing the filename of the saved EPR as a parameter to the tool. The tool will indicate an error condition if the user attempts to resume a resource that has not been previously suspended, or a resource that is done or has been destroyed.

Command syntax

```
globus-replication-resume [options]
```

Table 5. Options

-a,--anonymous	Use anonymous authentication. (requires either -m 'conv' or transport (https) security)
-c,--serverCertificate <file>	A file with server's certificate used for encryption. Used in the case of GSI Secure Message encryption
-d,--debug	Enables debug mode
-e,--eprFile <file>	Loads EPR from file
-f,--descriptor <file>	Sets client security descriptor. Overrides all other security settings
-g,--delegation <mode>	Performs delegation. Can be 'limited' or 'full'. (requires -m 'conv')
-h,--help	Displays help
-k,--key <name value>	Resource Key
-l,--contextLifetime <value>	Lifetime of context created for GSI Secure Conversation (requires -m 'conv')
-m,--securityMech <type>	Sets authentication mechanism: 'msg' (for GSI Secure Message), or 'conv' (for GSI Secure Conversation)
-p,--protection <type>	Sets protection level, can be 'sig' (for signature) can be 'enc' (for encryption)
-s,--service <url>	Service URL
-z,--authorization <type>	Sets authorization, can be 'self', 'host' or 'none'

Name

globus-replication-finditems -- This tool queries the replication resource to return the status of individual replication item activities.

`globus-replication-finditems`

Tool description

This tool provides the ability to query the status of individual replication items (e.g., replication of a specific file or files) managed by the given Replication resources. It is possible to query for the status of a specific named item or to query for the status of multiple items based on a particular status (e.g., Pending, Finished, Failed). In addition, to reduce potentially large overhead of returning a large results set to the client, the client may specify an offset and limit for the results set to be returned. The "name" or "status" option must be specified.

Command syntax

```
globus-replication-finditems [options] {-N name | -S status}
```

Table 6. Options

-a,--anonymous	Use anonymous authentication. (requires either -m 'conv' or transport (https) security)
-c,--serverCertificate <file>	A file with server's certificate used for encryption. Used in the case of GSI Secure Message encryption
-d,--debug	Enables debug mode
-e,--eprFile <file>	Loads EPR from file
-f,--descriptor <file>	Sets client security descriptor. Overrides all other security settings
-g,--delegation <mode>	Performs delegation. Can be 'limited' or 'full'. (requires -m 'conv')
-h,--help	Displays help
-k,--key <name value>	Resource Key
-l,--contextLifetime <value>	Lifetime of context created for GSI Secure Conversation (requires -m 'conv')
-L,--limit <num>	Limit on the size of the result set.
-m,--securityMech <type>	Sets authentication mechanism: 'msg' (for GSI Secure Message), or 'conv' (for GSI Secure Conversation)
-N,--byName <name>	Finds item by the Logical Filename (LFN) name.
-O,--offset <num>	Offset into the results set. Indexed by 0.
-p,--protection <type>	Sets protection level, can be 'sig' (for signature) can be 'enc' (for encryption)
-S,--byStatus <status>	Finds item(s) by status. Valid status values include "Pending", "Finished", "Failed", and "Terminated".
-s,--service <url>	Service URL
-z,--authorization <type>	Sets authorization, can be 'self', 'host' or 'none'