

# **GT 4.2.1 WS RLS : User's Guide**

---

## GT 4.2.1 WS RLS : User's Guide

### Introduction

The WS Replica Location Service (WS RLS) provides a Web services interface for the existing Replica Location Service (RLS). The features of the RLS have been preserved and interfaces should look familiar to users accustomed to working with the RLS. In most cases the interfaces are semantically similar (or identical) to the RLS operations, but effort has been made to adapt the interfaces to conform to expected style and conventions more typical of Web services. The interfaces consist of the WS-ReplicaLocationCatalog and the WS-ReplicaLocationIndex, representing the RLS Local Replica Catalog (LRC) and RLS Replica Location Index (RLI).



### Note

Please note this is relatively new code and may not necessarily be as robust as the rest of RLS.

Usage of the WS-RLS is related to but not exactly similar to the RLS. Users of the RLS will find that RLS operations have been organized into a set of command-line tools. The command-line tools provide capabilities for updating replica catalogs. For instance, a tool (globus-replicolocation-createmappings) supports the creating of new logical name to target name mappings. Such tools exist for creating mappings, adding mappings, deleting mappings, defining attributes, undefining attributes, and adding attributes, modifying attributes, and removing attributes. For query operations, both against the catalog as well as the index service, the WS-RLS supports the wsrf-query command with a custom query dialect (see usage for examples of the query dialect).

---

# Table of Contents

1. Mapping Replica Locations (WS) .....	1
1. Create mappings .....	1
2. Add mappings .....	1
3. Define attribute definitions .....	1
4. Add attributes .....	1
5. Query mappings .....	2
6. Query attributes .....	2
I. WS RLS Commands .....	?
globus-repicalocation-createmappings .....	4
globus-repicalocation-addmappings .....	5
globus-repicalocation-deletemappings .....	6
globus-repicalocation-defineattributes .....	7
globus-repicalocation-undefineattributes .....	8
globus-repicalocation-addattributes .....	9
globus-repicalocation-modifyattributes .....	10
globus-repicalocation-removeattributes .....	11
2. Troubleshooting .....	12
1. Errors .....	12
Index .....	13

---

## List of Tables

1. globus-repicalocation-createmappings Options .....	4
2. globus-repicalocation-addmappings Options .....	5
3. globus-repicalocation-deletemappings Options .....	6
4. globus-repicalocation-defineattributes Options .....	7
5. globus-repicalocation-undefineattributes Options .....	8
6. globus-repicalocation-addattributes Options .....	9
7. globus-repicalocation-modifyattributes Options .....	10
8. globus-repicalocation-removeattributes Options .....	11
2.1. WS Replica Location Service (WS RLS) Errors .....	12

---

# Chapter 1. Mapping Replica Locations (WS)

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

## 1. Create mappings

Use the `globus-replicolocation-createmappings(1)` tool to create mappings.

```
% $GLOBUS_LOCATION/bin/globus-replicolocation-createmappings \  
-s https://localhost:8443/wsrp/services/ReplicaLocationCatalogService \  
mydata1 gsiftp://path/a/to/mydata1
```

No output is expected from this command when successful.

## 2. Add mappings

Use the `globus-replicolocation-addmappings(1)` tool to add mappings.

```
% $GLOBUS_LOCATION/bin/globus-replicolocation-addmappings \  
-s https://localhost:8443/wsrp/services/ReplicaLocationCatalogService \  
mydata1 gsiftp://path/b/to/mydata1
```

No output is expected from this command when successful.

## 3. Define attribute definitions

Use the `globus-replicolocation-defineattributes(1)` tool to define attribute definitions.

```
% $GLOBUS_LOCATION/bin/globus-replicolocation-defineattributes \  
-s https://localhost:8443/wsrp/services/ReplicaLocationCatalogService \  
myattr1 logical string
```

No output is expected from this command when successful.

## 4. Add attributes

Use the `globus-replicolocation-addattributes(1)` tool to add attributes.

```
% $GLOBUS_LOCATION/bin/globus-replicolocation-addattributes \  
-s https://localhost:8443/wsrp/services/ReplicaLocationCatalogService \  
myattr1 logical string
```

```
mydata1 myattr1 logical string attribute-value-goes-here
```

No output is expect from this command when successful.

## 5. Query mappings

Use the [wsrf-query](#) tool to query mappings.

```
% $GLOBUS_LOCATION/bin/wsrf-query \  
-s https://localhost:8443/wsrf/services/ReplicaLocationCatalogService \  
"query-target: mydata1" \  
"http://globus.org/replica/location/06/01/QueryDialect" \  
<ns1:MappingStatusType ns1:logical="mydata1" \  
ns1:target="gsiftp://path/a/to/mydata1" \  
xmlns:ns1="http://www.globus.org/namespaces/2005/08/replica/location"/> \  
<ns1:MappingStatusType ns1:logical="mydata1" \  
ns1:target="gsiftp://path/b/to/mydata1" \  
xmlns:ns1="http://www.globus.org/namespaces/2005/08/replica/location"/>
```

## 6. Query attributes

Use the [wsrf-query](#) tool to query attributes.

```
% $GLOBUS_LOCATION/bin/wsrf-query \  
-s https://localhost:8443/wsrf/services/ReplicaLocationCatalogService \  
"query-logical-attributes: mydata1" \  
"http://globus.org/replica/location/06/01/QueryDialect" \  
<ns1:AttributeStatusType ns1:key="mydata1" ns1:name="myattr1" \  
ns1:objtype="logical" ns1:status="attributeExists" ns1:valtype="string" \  
xmlns:ns1="http://www.globus.org/namespaces/2005/08/replica/location"> \  
<_value xmlns="">attribute-value-goes-here</_value> \  
</ns1:AttributeStatusType>
```

---

# WS RLS Commands

The WS RLS provides a set of command-line tools to create, add, remove mappings between logical names and target names, define and undefine attribute definitions, and create, modify, and delete attributes. 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.).

The WS RLS command-line tools make use of the Common Java Client Options. These options are referred to below as [options].

---

# Name

`globus-repicalocation-createmappings` -- This tool is used to create mappings between logical names and target names. The *create* semantic implies that the logical name does not exist at the time of invocation.

`globus-repicalocation-createmappings`

## Tool description

Use this tool to create mappings between logical names and target names in the replica location catalog. The mapping must not exist. In addition, the logical name must not exist.

## Command syntax

```
globus-repicalocation-createmappings [options] \  
{ { logical-name target-name }+ | input-file | - }
```

**Table 1. globus-repicalocation-createmappings Options**

<b>{ logical-name target-name }+</b>	A listing of logical name to target name mappings.
<b>input-file</b>	A file containing logical name to target name mappings.
<b>-</b>	Standard input stream containing logical name to target name mappings.

---

# Name

`globus-repicalocation-addmappings` -- This tool is used to add mappings between logical names and target names. The *add* semantic implies that the logical name does exist at the time of invocation.

`globus-repicalocation-addmappings`

## Tool description

Use this tool to add mappings between logical names and target names in the replica location catalog. The mapping must not exist. In addition, the logical name must exist.

## Command syntax

```
globus-repicalocation-addmappings [options] \  
{ { logical-name target-name }+ | input-file | - }
```

**Table 2. `globus-repicalocation-addmappings` Options**

<code>{ logical-name target-name }+</code>	A listing of logical name to target name mappings.
<code>input-file</code>	A file containing logical name to target name mappings.
<code>-</code>	Standard input stream containing logical name to target name mappings.

---

# Name

globus-repicalocation-deletemappings -- This tool is used to delete mappings between logical names and target names.

globus-repicalocation-deletemappings

## Tool description

Use this tool to delete mappings between logical names and target names in the replica location catalog. The mapping must exist.

## Command syntax

```
globus-repicalocation-deletemappings [options] \  
{ { logical-name target-name }+ | input-file | - }
```

**Table 3. globus-repicalocation-deletemappings Options**

{ logical-name target-name }+	A listing of logical name to target name mappings.
input-file	A file containing logical name to target name mappings.
-	Standard input stream containing logical name to target name mappings.

---

# Name

globus-repicalocation-defineattributes -- This tool is used to define attributes.

globus-repicalocation-defineattributes

## Tool description

Use this tool to define attributes. Attribute definitions must be given a name unique within the local instance of the replica location catalog. Attribute definitions must be given a value type of dateTime, decimal, integer, or string. And attribute definitions must be associated with an object type of logical or target.

## Command syntax

```
globus-repicalocation-defineattributes [options] \  
{ { name object-type value-type }+ | input-file | - }
```

**Table 4. globus-repicalocation-defineattributes Options**

<b>{ name object-type value-type }+</b>	A listing of attribute name, associated object-type, and value-type.
<b>input-file</b>	A file containing the listing of attribute name, associated object-type, and value-type.
<b>-</b>	Standard input stream containing the listing of attribute name, associated object-type, and value-type.

---

# Name

globus-repicalocation-undefineattributes -- This tool is used to undefine attributes.

globus-repicalocation-undefineattributes

## Tool description

Use this tool to undefine attributes. Attribute definitions must be identified by the definition's name and associated object-type. The operation will clear attribute values for existing attributes with the definition's name.

## Command syntax

```
globus-repicalocation-undefineattributes [options] \  
{ { name object-type }+ | input-file | - }
```

**Table 5. globus-repicalocation-undefineattributes Options**

<b>{ name object-type }+</b>	A listing of attribute name and associated object-type.
<b>input-file</b>	A file containing the listing of attribute name and associated object-type.
<b>-</b>	Standard input stream containing the listing of attribute name and associated object-type.

---

# Name

globus-repicalocation-addattributes -- This tool is used to add attributes.

globus-repicalocation-addattributes

## Tool description

Use this tool to add attributes associated with logical names or target names. A corresponding attribute definition must exist. The logical name or target name with which to associate the attribute must exist. There must not be an existing attribute of the same type for a given logical name or target name. When adding attributes, the following parameters are required. The logical name or target name, referred to as the key. The name of the attribute as defined by an existing attribute definition. An object-type of logical or target. A value-type corresponding to dateTime, decimal, integer, or string. And finally a value compatible with the value-type.

## Command syntax

```
globus-repicalocation-addattributes [options] \  
{ { key name object-type value-type value }+ | input-file | - }
```

**Table 6. globus-repicalocation-addattributes Options**

<b>{ key name object-type value-type value }+</b>	A listing of key, attribute name, associated object-type, value-type, and value.
<b>input-file</b>	A file containing the listing of key, attribute name, associated object-type, value-type, and value.
<b>-</b>	Standard input stream containing the listing of key, attribute name, associated object-type, value-type, and value.

---

# Name

globus-replication-modifyattributes -- This tool is used to modify attributes.

globus-replication-modifyattributes

## Tool description

Use this tool to modify attributes associated with logical names or target names. Mutability of attributes is limited only to the attribute's value. The corresponding attribute must exist.

## Command syntax

```
globus-replication-modifyattributes [options] \  
{ { key name object-type value-type value }+ | input-file | - }
```

**Table 7. globus-replication-modifyattributes Options**

<b>{ key name object-type value-type value }+</b>	A listing of key, attribute name, associated object-type, value-type, and value.
<b>input-file</b>	A file containing the listing of key, attribute name, associated object-type, value-type, and value.
<b>-</b>	Standard input stream containing the listing of key, attribute name, associated object-type, value-type, and value.

---

# Name

globus-repicalocation-removeattributes -- This tool is used to remove existing attributes.

globus-repicalocation-removeattributes

## Tool description

Use this tool to remove existing attributes associated with logical names or target names. The corresponding attribute must exist.

## Command syntax

```
globus-repicalocation-removeattributes [options] \  
{ { key name object-type }+ | input-file | - }
```

**Table 8. globus-repicalocation-removeattributes Options**

<b>{ key name object-type }+</b>	A listing of key, attribute name, and associated object-type.
<b>input-file</b>	A file containing the listing of key, attribute name, and associated object-type.
<b>-</b>	Standard input stream containing the listing of key, attribute name, and associated object-type.

---

# Chapter 2. Troubleshooting

For sys admin debugging information, see [Chapter 5, Debugging](#).

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

## 1. Errors

**Table 2.1. WS Replica Location Service (WS RLS) Errors**

Error Code	Definition	Possible Solutions
Error: java.lang.Null- PointerException	When invoking the WS RLS command-line clients, a system-level exception like the one above may be encountered.	The admin should check the container logs for the exact error.
Error: A server error occurred while processing the request	When invoking the WS RLS command-line clients, a server error like the one above may be encountered.	The admin should check the container logs for the exact error.
java.lang.UnsatisfiedLinkError	This exception when using the WS RLS may indicate that the native RLS libraries that WS RLS depends on could not be located.	To correct this problem, ensure that the \$GLOBUS_LOCATION/lib directory is in the library search path (on some systems this is the LD_LIBRARY_PATH variable).
Unable to connect to localhost:39281	The WS RLS is an interface layer that depends on the RLS for the replica location functionality. You must install and run RLS and configure WS RLS to use the RLS via its JNDI configuration.	Check that RLS is installed, running, and check that the WS RLS JNDI configuration uses the correct host-name and port to connect to the RLS.
org.globus.com-mon.ChainedIOException: Failed to initialize security context	If this exception occurs while using WS RLS, it may indicate that the user's proxy is invalid.	To correct the error, the user must properly initialize the user proxy. See <a href="#">grid-proxy-init</a> for more information on proxy initialization.
Error: org.xml.sax.SAXException: Unregistered type: class xxx	If this exception occurs when using the WS RLS, it may indicate that an Axis generated XML type, defined by the WS RLS XSD, was not properly registered. While all the XML types should get registered upon deployment without intervention by the user, sometimes they do not.	To remedy the situation add a typeMapping to the server-config.wsdd file under globus_wsrf_replicalocation_service. Use the format shown <a href="#">here</a> .

---

# Index

## A

- adding attributes, 9
- adding mappings, 5
- attributes
  - adding, 1
  - defining, 1
  - definitions, 1
  - querying, 2

## C

- creating mappings, 4

## D

- defining attributes, 7
- deleting mappings, 6

## E

- errors, 12

## M

- mapping
  - querying, 2
- mappings
  - adding, 1
  - creating, 1
  - querying, 2
- modifying attributes, 10

## R

- removing attributes, 11

## T

- troubleshooting
  - errors, 12

## U

- undefining attributes, 8